

GOMF!

version 3.0



the Gurubuster

GOMF! - The Manual

Revision 3.0

January 10, 1989

Hypertek/Silicon Springs' Telephone Consulting service is available to all Registered users to support your installation and ongoing operations at no charge. To become eligible for this service, you must complete and return the enclosed Software Registration Card.

The answers to almost all of your questions can be found in your manual. To save you time, please read it thoroughly before calling our Help Line. If your question concerns your hardware, contact your computer, disk drive, printer (etc.) manufacturer first. If your question concerns the program, we will be happy to receive your call at the following number during normal business hours (10 a.m. to 6 p.m. Pacific Standard Time).

Customer Support Help Line: (604) 524-1125

If you write to us, be sure to include a phone number where you can be reached during the day.

NOTE: Please read the Pre-Call Checklist, which you will find in Chapter 10 of this manual, before calling the Customer Support Help Line.

Thank you for purchasing this program! If you enjoy it, be sure to check out our other fine Amiga offerings, which include:

- The Deluxe MIDI interface, the most full-featured Music Instrument Digital Interface available for the Amiga 500/1000/2000.

CHAPTER 1 - INTRODUCTION 5

What is GOMF ? (7)

The Contents of your Package (11)

CHAPTER 2 - GETTING TO KNOW THE AMIGA .. 13

AmigaDOS and the CLI (13)

The Workbench and Intuition (13)

Quick Startup (15)

Hardware Installation (15)

Installing GOMF on your disks (15)

Installing GOMF on a Hard Disk (17)

Additional GOMF functions (18)

If you have a 68010 or 68020 (18)

Upgrade PATHS (19)

CHAPTER 3 - GETTING TO KNOW GOMF 21

New features of GOMF 3.0 (21)

The Update file (22)

CHAPTER 4 - USING GOMF3.0 25

The Tools (25)

What GOMF3.0 Does (25)

CLI Startup (25)

Workbench Setup (26)

Loading from the Workbench (28)

CHAPTER 5 - GETTING TO KNOW THE GURU .. 30

What Causes Errors? (30)

Error Types (30)

When a 'TASK HELD' occurs (31)

The Error Handler Display (32)

The Status Area (32)

The GOMF BUTTON (33)

Skipping over errors (33)

Thawing System freezes (34)

The gadgets (35)

GOMF (35)

WHAP (36)

BOOT (37)

GURU (37)
SCAT (37)

CHAPTER 6 - OTHER FUNCTIONS OF GOMF3.0 . 39

Using NUKE from the CLI (39)
NUKE from the Workbench (42)
The NUKE Task Selector (42)
The Recall Command (59)

CHAPTER 7 - EXAMPLES 47

The Errors Drawer (47)
Questions & Answers (49)

CHAPTER 8 - TUTORIAL SECTION 51

Beginner's Tutorial (51)
The Recall Command (70)
Advanced Tutorial Section (72)

CHAPTER 9 - ADVANCED APPLICATIONS 77

CHAPTER 10 - TROUBLESHOOTING 79

APPENDIX A - GURU MEDITATIONS 85

APPENDIX B - PUBLIC DOMAIN SOFTWARE ... 91

APPENDIX C - HARDWARE INSTALLATION ... 93

GLOSSARY 107

Chapter 1 - Introduction

DISCLAIMER: We like Amigas...lots. Really.

The Amiga is, as we all know, a multitasking computer. This in itself implies that more than one 'task' may be running on the machine. Unfortunately, as many (most? All?!) users discover, when two or more programs interact badly, the Amiga will do one of two things: a task may crash and be 'held', or the entire system goes down! (affectionately known as 'a visit from the Guru'.) The fact that the Amiga is a multitasking computer makes a Guru Meditation even more dreadful, for if just one program crashes, all your other programs and their memory-resident data can be wiped out when you are forced by the Guru to reboot your Amiga.

This is not to imply that the machine is inherently buggy, or doesn't work properly, or anything that may have been true at one time, but it does so happen that our friend Bob Loblaw, who spends A LOT of time hanging out at 'Amiga parties' was talking with a certain famous Amigoid who said something to the effect that "...when you got an error, and the 'SELECT CANCEL TO RESET' message came up, it was SUPPOSED to actually CANCEL, not crash!" Unfortunately, it seems that the guy who was writing that part of the operating system quit, and uh, well, it never got done...

About the time that Bob told us this, we were all wishing and waiting for a recoverable ramdisk for the Amiga, and Christian Johnsen, (one of our programmers who seemed to know many mysterious inner secrets about the machine) suggested that an alternative to a RAMDISK that would survive a crash would be to trap and dispose of those pesky errors, so that the machine wouldn't crash. Period. "Sure, Chris..." we scoffed.

Well, IMAGINE our surprise when he showed us the first version of GOMF. (GOMF1.0) And it darn well WORKED! It didn't trap Gurus (then) but it sure trapped those nasty TASK HELD messages and what bliss to select CANCEL and THEY DID!! We were beside ourselves with

glee, and the next thing we knew, Chris released the thing as Shareware, and RAVE reviews were pouring in from across the land. Info magazine called it a “welcome utility”, Commodore magazine glowingly reported “...without a doubt, the most important program this month [GOMF]...deserves a round of applause.” GOMF1.0 even made it onto “the Best of Fred Fish” collection.

We didn’t see Chris much for what seemed a very long time, but he kept telling us that GOMF2.0 would be much better, that it would have many more features, that it would trap virtually ALL errors. Well, 2.0 was finally released and Chris was heard to say, “What next?” Well...

We sat very still for a while and decided that the ultimate version would not only let you remove the program, but let you save the data instead of nuking it! After mumbling certain choice phrases, Chris set off again. When he came to my office and showed it to me, I must admit I was shocked to see he could actually do it. I nearly dropped dead when he recovered from a complete lockup!

So, my friends, here it is (finally). We're sorry it took so long, but, I'm sure you will agree it was worth the wait. You had better be sitting down the first time to try out the GOMF button, otherwise, someone may need to catch you. Enjoy!

- Dean Spyropoulos, Jan 10,1989

What is GOMF and what can it do for you?

GOMF is an advanced error-handling routine that effectively eliminates what every Amiga owner knows as “a visit from the Guru”. Please read the entire text of this chapter before using GOMF. It contains important information and could save you a lot of hassle.

The name is an acronym for ‘Get Outa My Face’. I’ve found myself mumbling this phrase all too often when public domain or even commercial programs have created errors, bringing on the ubiquitous guru meditation alert. Of course the majority of these system traps I’ve seen were caused by address errors in my own code, while in the early stages of writing and debugging my own software.

It is true that the user has the option, at the system requester presented before the alert, of simply ignoring it and continuing to run programs, even though the system is somewhat crippled. I, however, found this to be almost as inconvenient as rebooting. I did so only when I had to save valuable data to disk or was anticipating another problem with the program I was developing. At times I would run my program, select an option, crash, make notes, click the system requester to the rear, run the program again, select another option, crash, and make more notes, in a loop until I had all the information I could glean. Then I’d click a system requester back to the front and go through the guru meditation every Amiga user has been forced into. There had to be a better way!

Apparently in the nuclear power generation industry when the operating system senses a grievous enough malfunction it shuts itself down. The operators or technicians then say that the ‘system scrambled’. I think this is an appropriate way of describing the Amiga user’s predicament.

My first experiments with the system traps was to write an error handler module, which is linked to the assembler code I’m developing, that allows me to clean up and debug while never crashing at all. This worked rather well because the program that caused the error ‘knew’ itself intimately and could therefore extricate itself, upon user direction to do so, from the system, with no ill effects. I should mention at this point that the kind of crashes I’m talking about are only those that cause a ‘TASK

HELD' requester or GURU MEDITATION ALERT, not the ones that have the system so badly out of shape that the normal interface is gone (i.e. complete system lockup) The link module would protect system integrity only from it's own errors. This meant that, if some other utility I was using caused an error, then the guru would come with the handbasket.

The second version of GOMF was designed to provide its error handling features globally within the system. It would then respond to errors in any task or process. The error handler would know nothing about a program, except what it could fetch out of the system structures. This means that GOMF can only remove public memory, including, the task or process, its memory allocations, messages unanswered and display screens and windows, from the system. This works well. It may be that there is one more opener than closer left in a library, device or resource, but this doesn't impede normal operations like the Guru Alert is want to do. Any memory allocated and maintained in a private memory list or messages saved internally by a program, cannot be released to the system.

The new software/hardware version is much improved, in that GOMF can now 'skip' over errors and save valuable data, or release the machine from a total lockup. Additionally, you do not have to keep track of which program bombs, because GOMF will do it for you.

The benefits of the GOMF system make it worth using, I believe, even for the casual user. The most obvious reason is the reduction in the frequency of having to reboot the system at the Guru Meditation prompt. This will save you time. When you don't have to reboot, you don't have to wait while your startup-sequence configures your machine. I've found this a real boon because I was reticent to have a lot of background tasks or utilities activated, mainly because I'd have to wait for them to be set up each time I was forced to reboot. When using GOMF this is reduced to the minimum possible, I believe.

It should be pointed out that GOMF in NO WAY forces you to use a crippled, or otherwise corrupted RAM workspace. It does NOT force you to do anything. You merely have additional options besides crashing. If you want to, you may reboot, go to the GURU alert, remove the error-causing task, etc. If you don't have to reset the Amiga, you lose no data, unless the creator of the error has trashed your workspace. This saves your

work. You have the option of saving out to disk any important files. I suggest doing this to temporary disk files so that if the data is corrupted, you have a chance to verify this before overwriting your originals.

A recoverable ram disk, otherwise known as VD0:, (HIGHLY recommended Shareware from ASDG's Perry Kivolowitz) also protects contents of ram from losses caused by resets. They can be mounted upon rebooting to recover their previous contents. They will not protect against data corruption caused by a runaway program either, however. The main difference between GOMF and a recoverable ram disk, to a user or programmer, is the same as the main difference between VD0: and RAM:, other than recoverability. Floppy disk is reasonably fast. RAM: is very fast. VD0: is somewhere in between. From the standpoint of speed, GOMF and RAM: are preferable to VD0:. 'I feel the need....the need for speed!'

Another useful tool to have in your software arsenal, especially if you are a programmer, or use the CLI frequently, is PopCLI. This Public domain program is widely available on various Amiga user Group disks, and will allow you to 'pop' up a new CLI after GOMF has trapped a GURU error. Instead of being halted dead in his or her tracks, the intrepid hacker can now peruse GOMF's plain english description of the cause of the error, and then invoke a new CLI task such as a debugger to further examine (and hopefully eradicate) the problem-causing code!

I can hear some of you devious (intelligent?) computerists out there thinking... 'Why not use RAM: for speed, when disk intensive operations are called for, and place all important data in VD0: files so that they can be recovered if a reset is necessary, just in case GOMF is unable to recover from the crash?' You know, I'm glad you thought of that. It works well. GOMF, RAM: and VD0: are foundational tools for programming, at least in the early stages. Development is quicker and more reliable.

The GOMF1.0 software, listed below, may be shared with anyone, provided this is done for no profit, other than a nominal fee for disk copying. I require also that all files listed be kept together as found. I wish none of this text file nor any of the executable programs be altered in any way. GOMF1.0 and GOMF1.0.obj are shareware products which are to

be utilized only by those who pay the suggested sum of \$5.00 each for the use of either program. If you find this software useful, please remit this minimal sum to the address below.

Christian Johnsen

3169 Consort Court

Clearbrook

British Columbia

Canada

V2T 4J5

(604) 853-5426

GOMF1.0 is not public domain software. Therefore, neither GOMF1.0 nor GOMF1.0.obj may be included with any other program released, whether public domain or commercial without the prior, written consent of the author, Christian Johnsen. If you wish to do so please arrange this by writing me at the following address with the particulars. I'll entertain customizing this system for your particular application's use if required.

GOMF3.0 is also not Public Domain. However, unlike GOMF1.0, neither is GOMF3.0 Shareware or otherwise freely distributable. This is our 'bread and butter' version. Please help us make it a success and DON'T GIVE IT AWAY.

We wish to mention, at this point, that we hope no one is offended by the icon we've designed for these programs. No disrespect was intended.

The Contents of your GOMF Package:

The disk included in this package should contain the following programs and text files plus their attendant icons.

In the GOMF1.0 Drawer:

1. GOMFinfo - an abbreviated text file of instructions and terms pertaining to GOMF1.0.

2. GOMF1.0 - the Shareware global error handler.

3. GOMF1.0.obj - the error handler object code link module.

4. Hey! - a program to allow recall of GOMF1.0 for user discretionary operations on the program display.

5. Err1 - a simple Workbench screen error test program.

6. Err2 - an error test program with a custom double length screen, containing fourteen windows.

7. Err3 - a custom double length screen error test program containing no windows.

8. Err4 - an example test program linked with Gomf1.0.obj to demonstrate the capabilities of the link module.

9. Err4.asm - the source code for Err4 that demonstrates the use of the link module Gomf1.0.obj.

In the GOMF3.0 Drawer:

1. GOMF3.0 - The long-awaited blockbuster sequel to GOMF2.0!

2. Nuke - The feature we almost left out, NUKE can remove ANY task, window, or device from the system. USE WITH CARE!

3. Recall - This command calls GOMF, to allow you to WHAP windows, etc.

4. Updates - A text file detailing updates, if any.

In the Errors Drawer:

5. Error1 - a simple Workbench screen error test program.

6. Error2 - an error test program with a custom double length screen, containing fourteen windows.

7. Error3 - a custom double length screen error test program containing no windows.

8. Error4 - a busy loop (endless loop) program that is useful to test the power of the NUKE command of GOMF3.0.

9. Error5 - a test program designed to lock up the Amiga.

10. Error6 - a program that locks up the Amiga while attempting to load Error1 (and leaves both drive lights on).

The Manual: The tome you are now reading.

Software Registration Card: The important card YOU WILL send in, right?

Chapter 2 - Getting to know the Amiga

AmigaDOS and the CLI

The CLI (figure 1.) is best described as the 'native mode' of the Amiga. It is the environment always present every time the machine boots up (You can break to CLI from a startup-sequence with a CTRL D) The CLI runs programs in one of two ways. Either they are started as 'background tasks' via the RUN command, or each task can be started from its own CLI via the NewCLI command.

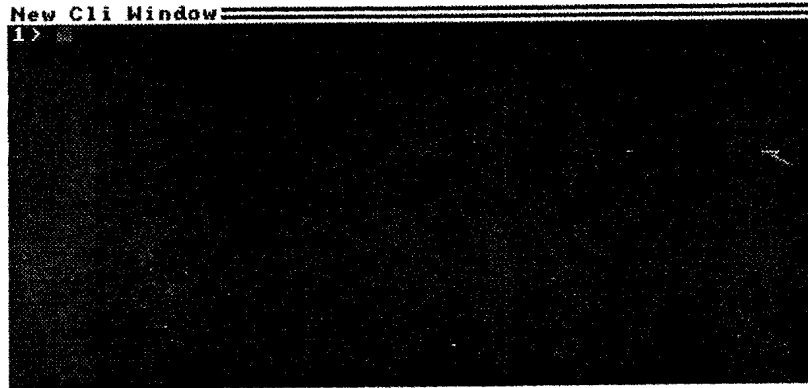


fig.1

The Workbench and Intuition

For inexperienced users, or those folks who appreciate the term "user-friendly" the Amiga's Workbench (figure 2.) is a blessing. It appears that (back in '85) when Commodore first decided on how to effectively (?) market the Amiga, they chose to completely focus on the

Workbench approach to using the machine. In fact, the user's guide supplied with the AMIGA 1000's contained almost no information whatsoever on correct usage of the Command Line Interpreter (CLI). The somewhat arcane language of the CLI was undoubtedly deemed too daunting for the average user by the Commodore-Amiga marketing department.

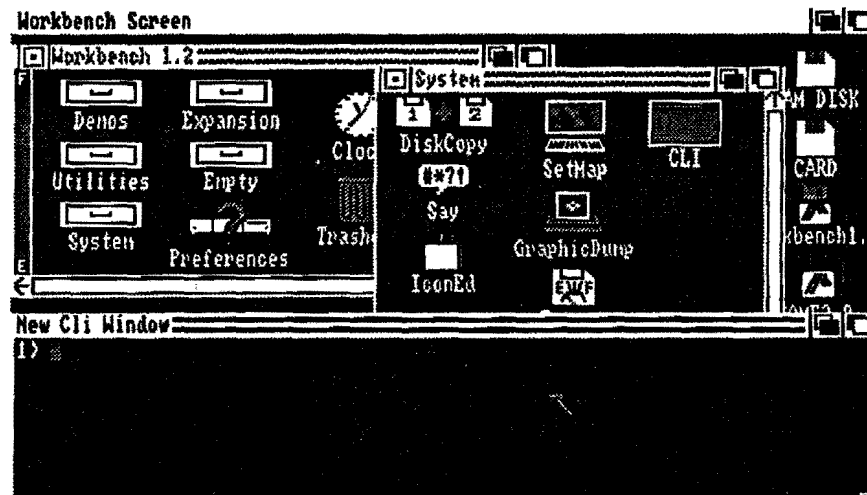


fig.2

The Workbench and the Amiga's windowing environment portion of the operating system (known as Intuition) have some important differences from the AmigaDOS CLI in the way that programs are run. As this relates to GOMF, the differences can be summarized as follows:

- Programs run from the Workbench are known as Processes.
- Multiple Workbench processes are roughly equivalent to opening a NewCLI for every task.

Quick Startup

For those of you who wish to get GOMFing RIGHT AWAY, simply insert the GOMF disk, and double-click the GOMF3.0 icon to install GOMF. Then refer to the Beginner's Tutorial, in Chapter 7. For best results, however, you should perform the installation procedures described below. If the installation process should prove too difficult for you, ask your dealer or other AMIGA owner to assist you in putting GOMF3.0 in your Workbench's Startup-Sequence.

Hardware Installation

If the version you have purchased has the "GOMF BUTTON" included in the package, please refer to Appendix C for information regarding installation.

Installing GOMF on your own disks

What you will require:

- One or more (Workbench 1.2 or CLI) bootable startup disks.

NOTE: The installation procedure described below should be performed on a BACKUP of your original Workbench or application disk. This will safeguard you against accidents, should something go awry. Please refer to the User's guide that came with your Amiga for details on this operation.

To start the program from the CLI under Workbench 1.2 enter....

RUNBACK GOMF3.0

We highly recommend using Workbench 1.3 but until then, use RUNBACK. You must copy RUNBACK to the C: directory of your system disk. We have enclosed a copy of this Public Domain program free of charge for your convenience. It can be found in the PD drawer of your GOMF3.0 disk.

If you have Workbench 1.3, you must RUN GOMF, add this

```
STACK 2000  
RUN > NIL: GOMF3.0  
STACK 4000
```

and you will be able to close your CLI. The command STACK 2000 reduces the stack assigned to the GOMF task, to the minimum safely advisable, and reduces the memory consumption required by GOMF from the system. The command STACK 4000 is the current default set by AmigaDOS. If you are using programs that use large chunks of memory and are crashing, try specifying a larger stack. GOMF does not actually need this command, however, if you have problems, simply remove them from your startup.

Place GOMF3.0 as early in your startup-sequence as possible. This is to allow it to capture a valid snapshot of the Trap and Exception vector tables. That is, before any tasks or processes have had an opportunity to step on them.

The usual method for modifying a Startup-Sequence is as follows:

1. Boot with a write-enabled backup of your Workbench 1.2 disk.
2. Double-click on the System drawer icon.
3. Double-click on the CLI icon.
4. Type the following commands:

Ed Startup-Sequence

5. At or near the beginning of the Startup-Sequence, add the lines as described above.

6. Tap the ESC key, then press X <Return> to write the newly-modified Startup-Sequence to your Workbench backup.

7. Now, insert the GOMF disk, and drag the GOMF3.0 icon (Inside the GOMF3.0 drawer) to the main directory of your Workbench Backup. Obviously, single drive users will be required to swap disks. If you intend to use GOMF's tools (NUKE and RECALL), transfer these as well.

8. If your Workbench disk is too full, you may have to delete some files to make room for GOMF3.0. The Demo drawer is a good candidate for deletion. Single-click on the Demos drawer, then hold the right button and select DISCARD from the project menu.

Click with the left mouse button on OK TO DISCARD to delete the Demos drawer and its contents.

Installing GOMF on a Hard Disk

You will require:

- A backup (Workbench 1.2) of your Hard Drive "BOOT DISK"

The steps for installing GOMF3.0 on your hard drive are much the same as for a floppy-based system, except that the hard drive Startup code must be executed before GOMF can be accessed from the hard drive. Usually this is accomplished with the command

BindDrivers

and in the case of A1000/Sidecar and A2000/Bridgecard-type hard drives,

DJMount

Other hard drives may vary, refer to your Hard disk documentation for details. As described earlier, place GOMF3.0 as early as possible in the Startup-Sequence for maximum efficiency.

ADDITIONAL GOMF STARTUP FUNCTIONS

GOMF also has two additional features which can be enabled from your startup-sequence. If you type

```
RUN > NIL: GOMF3.0 ^
```

then GOMF will automatically protect your low memory autovectors. For example, MARAUDER is a piece of software which will trash these memory locations. If you have vector checking turned on at the time, GOMF will come up to tell you that your memory location has been violated and will ask you to remove MARAUDER. You will have to disable this vector checking ability before running MARAUDER. Please see the section NUKE for more information about this.

IF YOU HAVE A 68010 or 68020 PROCESSOR

There is an instruction for the 68000 processor (the one that comes standard with your AMIGA) that when run on either the 68010 or 68020 causes a guru. This will not happen if you have the 68000.

If you should happen to have either of these processors, you should alter your startup sequence as follows:

```
RUN > NIL: GOMF3.0 &
```

The '&' symbol specifies that GOMF will not allow this to happen. Both the '&' and the '^' functions can be activated simultaneously or individually. It doesn't matter which one appears first, as long as there is a space between them, as follows:

RUN > NIL: GOMF3.0 & ^

For those interested in the precise workings of the '&' function, read on. The 68010/20 machines, issuing a MOVE SR,<ea> will cause a privilege violation. This switch enables the GOMF feature that senses the violation, the instruction is parsed, and if it is indeed this violation, the MOVE SR,<ea> opcode is converted in memory to the equivalent legal instruction MOVE CCR,<ea>. This occurs transparently to the user, and it is patched instantly in memory. Any other type of privilege violation is handled by GOMF in its' normal way. ONLY this specific error on 010/20 machines is patched with the '&' function.

The preceding functions can also be activated from the Workbench. Please see the tutorial section for details.

UPGRADE PATHS

IF YOU ARE UPGRADING FROM AN EARLIER VERSION

If you have followed the above instructions for installing GOMF and are upgrading from GOMF 2.0/2.1/2.2 then you must also use the new tools that are provided with GOMF 3.0. The NUKE and RECALL tools supplied with previous versions of GOMF are incompatible with GOMF 3.0. In an effort to provide a reliable and efficient program, it became necessary to alter the way in which GOMF would communicate with its tools. When you replace your old version with the new one, please be sure to replace these tools as well.

IF YOU WISH TO UPGRADE TO THE GOMF BUTTON

If you have purchased the GOMF3.0 software and wish to have

a GOMF BUTTON, please contact us. This upgrade will be available only from us.

To order, write to us at:

812 Surrey Street
New Westminster, B.C. Canada
V3L 4W2

Or order TOLL FREE 1-800-663-8526 (US ONLY)
(604)-524-1125 in Canada

Chapter 3 - GETTING TO KNOW GOMF

GOMF3.0 is a MUCH IMPROVED AND ENHANCED version of the original Shareware (GOMF1.0) release. Like the original, GOMF3.0 is a global system error handler designed for the average Amigo. It is NOT intended as an advanced tool "just for programmers", although it is highly appropriate for such usage.

(There is another version of the program, Gomf1.0.obj, which IS designed for programmers. It must be linked to a program after assembly or compilation. It will handle errors similar to GOMF1.0, except that it can be customized as described in 'Using Gomf1.0.obj'. It is intended for non-commercial use ONLY.)

New features of GOMF 3.0

The program is now much more powerful and has been assigned a full version number increment. The first two features are available only if you have purchased GOMF3.0 with the GOMF BUTTON (Hardware/Software version) To upgrade your GOMF3.0 Software version to the Hardware/Software version, please refer to the section entitled 'Upgrade Paths'.

1. By means of the GOMF BUTTON hardware device, you can actually 'skip' over errors. When a program has crashed, the GOMF requester appears. If you have the GOMF BUTTON installed, a simple push of a button will readdress your program at a different memory location and survive the crash. This allows you to save your important data contained within the application that caused the error. The addition of this feature was the most frequently requested improvement to GOMF by end-users.

2. The next-most common request was to escape a total system lockup. This was a tall order, but we succeeded (conditionally). The state

of the system under these circumstances is largely unknown. Because of this fact, we do not claim that you will escape EVERY one of these states. An example program is provided for your testing.

3. GOMF3.0 now allows you to turn Vector Table checking off or on. Some nasty programs would alter these Tables a bit (without really damaging anything) and the GOMF window would still come up and tell you that there was an error, just the same. The default is to ignore the Vector Tables. Please see Chapter 5 for more information.

4. This new version of GOMF does not require the use of the 'whap' gadget for removing windows or screens, in most cases. GOMF is now smart enough to know in which window or screen the error has occurred, providing that GOMF was resident when the window or screen was created by the system.

5. GOMF3.0 now handles the supervisor stack frame correctly for the 68010 and 68020 microprocessors. However, you may not use DeciGel. To make life problem-free, use GOMF3.0 instead.

6. The GOMF start-up window is much speedier than before.

7. There seems to be a confirmed problem with anyone who has more than 512K memory in an AMIGA addressed at \$C00000. Workbench 1.3 has a program called SETPATCH which will improve reliability with this extra memory.

8. GOMF has been modified to support more of the diverse display possibilities of the Amiga, such as PAL video and MOREROWS.

9. EVERY capability of GOMF has been enhanced where required to increase reliability.

The Update file:

On your program disk, you may find a file entitled 'Updates'. This

file will contain information about any new features added since this manual was printed. From time to time, the version number of GOMF may change, as well. (i.e. GOMF 3.01)

Chapter 4 - Using GOMF3.0:

Great care must be taken in the using of these tools. As with practically any tool, it can be used destructively as well as constructively. Use them wisely. Practice with the error creating programs supplied until you have an intimate understanding of how the GOMF3.0, Nuke and Recall commands work together. You must use these tools with care to effect maximum benefit.

What GOMF3.0 Does:

It notifies the user that it has installed itself, then snoozes until the system scrams. Then, according to user selection, it restores the system in one of four ways.

The program can be run either from the CLI or the Workbench.

CLI Startup

For further details of GOMF options, refer to the section entitled "Installing GOMF on your own disks."

To start the program from the CLI under Workbench 1.3 enter...

RUN > NIL: GOMF3.0

Note that if you do not RUN GOMF3.0 under workbench 1.3 you will not be able to close the initial CLI window.

To start the program from the CLI under Workbench 1.2 enter....

RUNBACK GOMF3.0

We highly recommend using Workbench 1.3 but until then, use RUNBACK. You must copy RUNBACK to the C: directory of your system disk. We have enclosed a copy of this Public Domain program free of charge for your convenience. It can be found in the PD drawer of your GOMF3.0 disk.

We suggest that GOMF3.0 also be kept in the C directory of the Workbench disk, and the above line be added to your startup-sequence. This way your Amiga system will be made reasonably crash proof upon booting up.

Workbench Setup

As with startup from the CLI, two additional options can be enabled from the workbench.

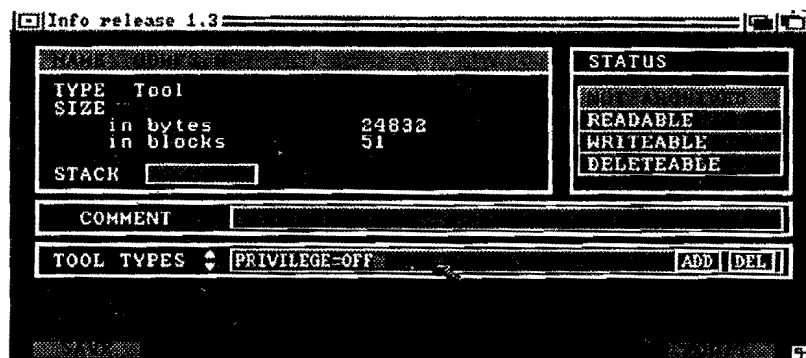
GOMF also has two additional features which can be enabled by altering the icon which you normally double-click on.

GOMF has the ability to automatically protect your low memory autovectors. For example, MARAUDER is a piece of software which will trash these memory locations. If you have vector checking turned on at the time, GOMF will come up to tell you that your memory location has been violated and will ask you to remove MARAUDER. You will have to disable this vector checking ability before running MARAUDER.

There is an instruction for the 68000 processor (the one that comes standard with your AMIGA) that when run on either the 68010 or 68020 causes a guru. This feature repairs this problem, should you have either of these processors.

Both the vector-checking and the 68010/20 fix, described above, can be activated simultaneously or individually. The activation of these features is accomplished by the following procedure:

1. Load Kickstart and Workbench.
2. Insert the GOMF3.0 disk and open it by double-clicking.
3. Click ONCE on the GOMF3.0 icon, then move the cursor to the workbench menu bar. Press the right mouse button and select "info" from under the Workbench heading and release the button.
4. The info program should now be loaded.



5. Refer to the section marked TOOLTYPES. The phrase VECTORS=OFF will be displayed in the entry field. If you want to enable vector checking, position the cursor to the entry field and edit 'OFF' to 'ON'. It is not necessary to hit return.
6. Click on the 'down-arrow gadget' of TOOLTYPES. The display will now reveal PRIVILEGE=OFF. Likewise, to enable this feature, edit the 'OFF' to 'ON'.
7. VERY IMPORTANT! After you have made the changes to your

satisfaction, select the SAVE gadget at the lower left of your screen. Following this procedure, each time you run GOMF from the workbench, your custom default features will be activated.

If you change the tooltypes, running GOMF from the CLI will not be affected by these changes.

You should also note that by editing the icon and selecting your defaults will only affect THAT particular icon. If you have multiple copies of GOMF, make certain that the icon used is the one with your selected tooltypes.

If you need more information about tooltypes, please refer to the section in your AMIGA manual entitled "Workbench Tools".

LOADING FROM THE WORKBENCH

When you double-click on the GOMF3.0 icon there will be a momentary window displayed informing you that GOMF3.0 is activated (figure 3). It will also tell you if any of the special features have been activated. It will detect any setup errors and report these, if found, so that you'll know that the system is then not protected. GOMF3.0 (and Gomf1.0.obj) examine system structures to determine which of either 68000 or 68010/20 microprocessors are on board and configures itself to handle the appropriate supervisor stack frame.

GOMF 3.0 'the Gurubuster' © 1988

© Hypertek/Silicon Springs

**This version of GOMF is not a public
domain, nor shareware, release.**

**Please respect this
notice of copyright.**

fig.3

Chapter 5 -

Getting to know the GURU

What Causes Errors?

When a piece of software (or hardware) running on the Amiga fails due to a bug in the program, or an anomaly in the way it interacts with the rest of the system, an error condition is created.

Error Types

There are three major classes of error types on the AMIGA. The first is shown below, in figure 4.

SOFTWARE ERROR-TASK HELD
Finish ALL Disk Activity
Select Cancel to reset-debug

fig.4

This type of message signifies a 'partial crash' that is not usually disastrous to the rest of the system UNTIL you select CANCEL. The second is the dreaded GURU MEDITATION ALERT, (figure 5) which until now was typically instantly fatal.

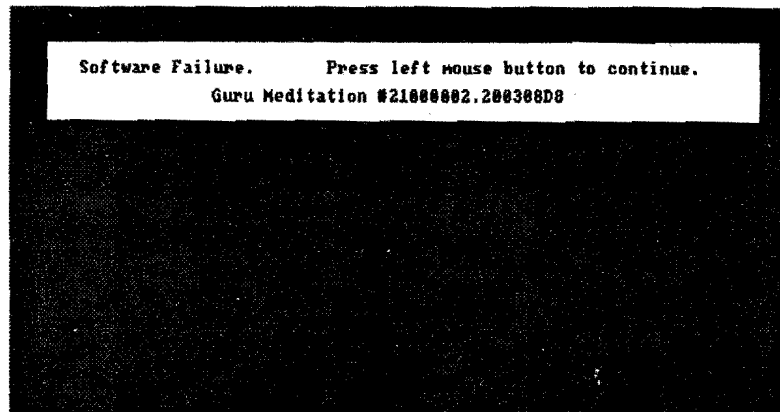


fig.5

The third is a catastrophic system failure characterized by an instant lockup of the entire system. Until now, this 'sudden death' was fatal to ALL programs. GOMF3.0 with the GOMF BUTTON hardware has the ability to recover from some of these types of errors. Luckily, programmers so inept as to write software that does this seldom get their work published, and as such, this condition is very rarely encountered (hopefully).

WHEN A 'TASK HELD' ERROR OCCURS:

Once an error is encountered, the normal system requester, giving the RETRY or CANCEL options will be presented. Select the CANCEL gadget. The GOMF 3.0 'The Error Handler' window (figure 6.) will then be presented.

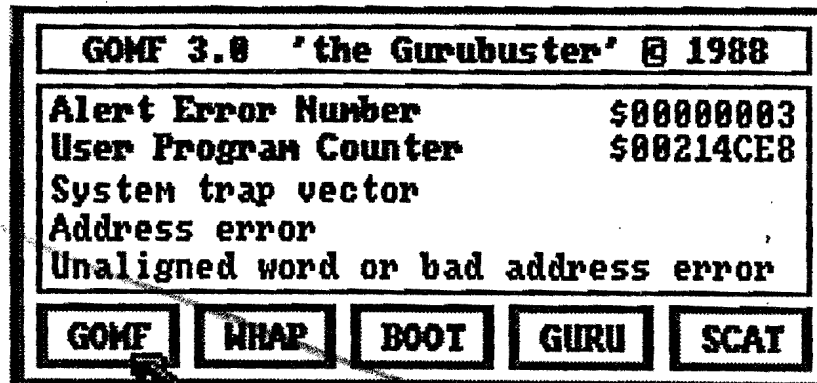


fig.6

The Error Handler Display

This window contains useful information, similar to that encoded into the guru meditation alert number, but it is decoded and presented in English text. This includes the alert number, the program counter address of the instruction immediately following the error, the library or vector type of the error, the general type of problem and a description of a more specific nature. This includes all known error types listed in the developer's alerts include file as well as Paged Memory Management units and Floating Point Co-Processor trap errors. There are also five gadget options. You must use caution when using any of this tool's options.

The Status Area

Refer to Appendix A for a list of Guru Meditation Alert numbers and their meanings.

PLEASE NOTE: If some of this information is too technical for you, please don't worry! It has been included, in addition to the English text, for the benefit of the curious folks out there.

THE GOMF BUTTON

The GOMF BUTTON hardware will allow you to access two features not supported by software. The software is the same whether you have the hardware or not, however, you must have the hardware to skip errors and recover from lockups.

SKIPPING OVER ERRORS

When a program gurus (either the task-held requester type or the direct guru) and you are presented with the GOMF requester, you now have the option of skipping the error and continuing execution of the program. This is accomplished by simply pressing the GOMF BUTTON.

This means that the data and your valuable efforts have not necessarily been lost. After pressing the button and the requester has disappeared, the program can still be used. We recommend that you now save your data in a temporary file. Before replacing any previous versions of this file, be certain to satisfy yourself that the data in this file is intact.

If the requester reappears after the error is skipped, the program is probably no longer viable.

MULTIPLE SKIPS

If you simply cannot believe that the program will not come back, you can try up to ten times to skip over the error. After that, a window will appear advising you to use the GOMF gadget to remove the program.

However, if you are of the die-hard opinion that your program still has hope, then you are given the option to wait ten seconds before trying another ten times to skip the error. Be aware that while attempting to skip beyond ten tries, you are treading on thin ice.

ERRORS THAT CAN'T BE SKIPPED

There are GOMF messages that will advise you of certain conditions that dictate that the program cannot be restored. If the error has occurred:

- A) during the execution of a ROM routine
- B) when the Program Counter is pointed at a ROM routine
- C) when the Program Counter is pointed at any of the custom chip addresses
- D) when the Program Counter is pointed at low memory system vectors

then GOMF prevents you from attempting to skip the error, because this greatly improves the reliability of the function.

THAWING SYSTEM FREEZES

There may be times that your AMIGA decides to lockup and die. Normally, this would mean a total reset of your machine. [I just HATE IT when that happens]. With the GOMF BUTTON hardware installed, you now have some protection against this revolting development.

GOMF knows when the system is locked up or when a skip is possible. So, when the system is frozen and you press the GOMF BUTTON (as you would to skip an error) GOMF will unlock the system, if at all possible.

To test these functions of GOMF3.0, please refer to Chapter 6.

THE GADGETS:

GOMF

The GOMF gadget removes, as safely as possible, the errant program, then returns the system to normal processing. It will not fix the problem in the program, only remove the task or process without the need to reset the computer. Whenever the GOMF gadget is tweaked, the results of the operation are displayed. This would be either the completed message, if successful, or an error message in the event of a failure. Normally it is only the removal of the display elements of a program which yield errors. If GOMF3.0 has been running since startup, it keeps track of what windows and screens belong to what task or process. Selecting the GOMF gadget will effectively remove the display automatically. Should you absolutely need to run another program, such as a debugger, at this point, DO NOT select the GOMF gadget, as this may remove the display of the debugger. Finish using the debugger (or other program) and exit it before using the GOMF gadget. If GOMF3.0 was loaded after the task or process, you will have to 'WHAP' the display elements as outlined below. You can usually use the WHAP gadget as suggested in the error message to recover the display should this occur, before attempting to use GOMF a second time. The messages provided by the error handler should be sufficient to guide you through the removal of a program, but please use the provided error examples to practice operations.

NOTE: Causing deliberate errors with the Error programs is not particularly recommended when running other 'real' programs. The severe errors caused by these bugs may interfere with normal operations.

WHAP

The WHAP gadget is provided for use when a program's windows or screen elude the automatic scan features of GOMF or if you loaded

GOMF after an error had already occurred. For instance, if it is ascertained that only the WB screen is active, then you will be requested to click on the WHAP gadget followed by the offensive program's window or screen. WHAP will remove the display element selected. WHAP will make it possible to remove piece by piece a multi-screen and/or multi-window display, if necessary. You will have to either, use the Left-AMIGA and N or Left-AMIGA M key combinations to flip between the Workbench and the applications' screen if it uses one of its' own, or pull down the Workbench screen to expose the faulty programs' display, when using this option.

How to use WHAP:

When you click on GOMF's WHAP gadget, you have 2.5 seconds to click on the window that you wish removed from the display. WHAP will then remove that window's display elements. If you wish to WHAP a separate screen, then after clicking on WHAP, you MUST click on that screen's FRONT gadget (in the upper right corner.) If the screen does not have FRONT and BACK gadgets visible, then click in the upper right corner anyway. Very often, the gadgets are there, but invisible on a custom screen.

BOOT

The BOOT gadget allows a reboot of the system without visiting the guru or giving the Amiga three finger salute (also known as the Vulcan Nerve Pinch).

If you encounter a problem with a real time clock being set from within your startup-sequence, after a BOOT has been clicked upon, then the solution that works is to place the SetClock, or whatever your particular command is, near or at the end of the batch file. If necessary a Wait command with a duration of 5 seconds may also help. Any users who have other than the standard hardware configurations, such as hard disks or local area networks, will likely not find the BOOT gadget too helpful in resetting the hardware. This is because it is a quicker reboot that does not pull down to full reset ordinarily. If you have any problems with this

simply don't use the BOOT gadget but instead either use the GURU gadget and follow normal procedures, or issue the tried and true Amiga three finger salute.

GURU

The GURU gadget calls the normal system alert as usual.

SCAT

This gadget allows the movement of the GOMF3.0 requester from the top left corner of the display to the bottom right corner or back, if desired. This feature has been coded to be useful in both normal and interlace video modes.

Chapter 6 -

Other Functions of GOMF3.0

USING NUKE FROM THE CLI

Used in conjunction with GOMF3.0, the Nuke command allows the user to remove a program which may not have caused a system error, as yet, but which may have a functional error, such as an endless loop.

When the Nuke command has activated it, the GOMF3.0 requester display has the GOMF gadget replaced with the NUKE gadget, which functions in much the same way.

1. The purpose of the Nuke command is to remove a program with a Non-Guru Meditation Alert error. This is usually a functional or branching error within a program. Nuke can be used to remove such a program, or even a program that is in a normal state. Nuke'ing programs that can be terminated by other means is not recommended. If the close window gadget or quit menu option is available, then by all means use them.

2. The Nuke command, and the section of GOMF3.0 code that supports the Nuke function, have been coded to provide the best reliability possible, with the most memory freed to the system, reliability has, however, been given foremost priority.

3. Nuke can be used from both the CLI and the Workbench. This command is probably the most dangerous in the tool kit. Use it sparingly. Practice using it with the error programs or at least with a computer configuration which will not jeopardize any valuable data before using it in a 'real time' situation.

4. When Nuke is used from the CLI, the arguments are case sensitive. This allows multiple iterations of a program to be run under different case dependent names and permits the selective termination of the specific errant program with the Nuke command.

5. Only one argument is allowed at a time. The first program found in the Exec Ready Queue, Exec Waiting Queue or the AmigaDOS task array that matches the argument will be targeted for GOMF3.0 to eliminate. If no match is found then the program will display all available tasks and processes currently in the aforementioned lists. It should be noted that there are a few transient states which tasks can occupy which mean that the task is temporarily in none of these lists. Also, due to task switching and scheduling, a program may simply be between lists, in the process of being added to one while being removed from another. You may well find that one use of the Nuke command followed immediately by another displays significantly different results.

6. Nuke has the following syntax and adjectives.

Nuke

This causes a requester to be displayed. (See section entitled "The NUKE Task Selector" below.

Nuke ?

This form results in the command template being displayed.

Nuke !

This form of the Nuke command results in only the printing of the lists as mentioned in 5 above. No real Nuke is attempted. The command behaves as if it could not find a task victim to remove as described above.

Nuke <ProgramName>

This is the real thing. The program argument name is looked up in the system queues and GOMF3.0 is activated to cope with the removal of the named program.

Nuke ^

This command enables or disables vector-table checking. A mes-

sage will be returned informing you whether it has been turned on or off. The ^ symbol (called a caret) appears by holding down the shift key and pressing the number 6.

7. When used from the CLI Nuke outputs its lists and error messages to the size of the current CLI window. For optimum readability, we suggest that you size your CLI window as a full length window 1/3 of the screen width. This leaves a full 2/3 of the display for other purposes, such as the Workbench display. The command will print full length to either an interlaced or non-interlaced display.

8. If the display of the task lists must be paused during output then you may use the standard system methods. Use the space bar to pause and the backspace key to continue, or use the right mouse button to pause and release the right mouse button to continue.

9. When in the process of Nuke'ing a program, do not fiddle with any of the gadgets or menus of the program being Nuked, as this may result in a total lockup of the user interface.

10. If an error requester should pop up during a Nuke it should be ignored until after the Nuke operation is completed. Then, afterward, click on the CANCEL gadget and handle the error.

11. If an error occurs which goes directly to 'The Error Handler' requester during a Nuke, then all gadgets except WHAP are available as usual. If WHAP is clicked an error message will be displayed but a WHAP will not be generated. If a WHAP is required to clean up the display then you should use GOMF to return to normal processing and then use the Recall command to reactivate GOMF3.0 to facilitate the WHAP of display elements that must still be removed.

12. Nuke will now parse quoted, multi-word, arguments from the CLI. Should such a task exist-

Nuke "DF1:This Command"

would solve the problem.

13. Please be advised that the above mentioned system tasks, like "File System", may have unrecoverable effects upon the system if they are indeed Nuke'd. Devices can be Nuke'd. Be cautious with this command!

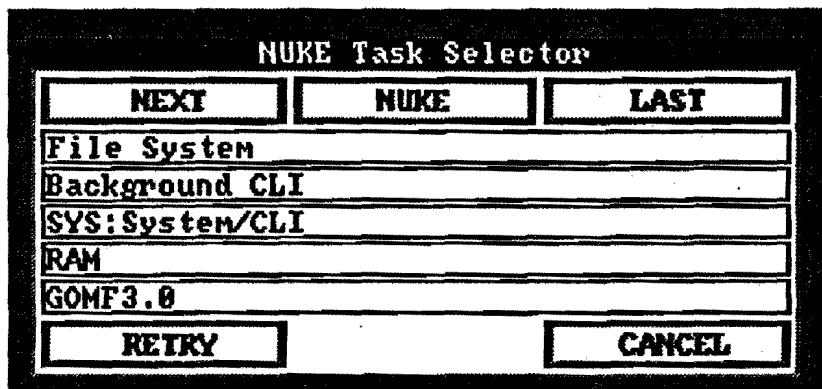
NUKE FROM THE WORKBENCH

14. Nuke can be used from the Workbench in one of two ways.

The first method used is called extended selection, as described in your AMIGA user's manual. To Nuke a program, run by double clicking on its icon, first press the shift key, hold it down through the next steps; click on the Nuke icon once, then select the icon of the program you wish removed and double click on it. You may now release the shift key. When the GOMF3.0 requester appears it will have the GOMF gadget replaced by the NUKE gadget. They are essentially the same in most operational respects.

THE NUKE TASK SELECTOR

The second method is initiated by double-clicking on the NUKE icon only, which will result in the NUKE Task Selector being displayed, as shown below.



The gadgets operate as follows:

TASK

FIELDS These fields contain the names of the tasks or processes currently in the Exec Task Ready Queue, followed by the Task Waiting Queue and AmigaDos Task Array. To select a task you want to NUKE, simply click on the gadget containing that name. This will cause that particular gadget to remain selected. When a task field gadget is selected in this manner, it will be removed by selecting the NUKE gadget. To deselect a task, click on it once again, or select any other gadgets.

NEXT This will display the next screen of five tasks. Occasionally, a blank field may appear. This cannot be selected.

LAST This gadget will display the previous five tasks.

NUKE Selecting NUKE causes the currently selected task or process to be eliminated. Only one task can be selected, and subsequently Nuke'd at a time. If no task is selected, nothing will be Nuke'd.

RETRY The purpose of RETRY is to reload the task field gadgets. It should be noted that there are a few transient states which tasks can occupy which mean that the task is temporarily in none of the system lists. Also, due to task switching and scheduling, a program may simply be between lists, in the process of being added to one while being removed from another. You may well find that one use of the Nuke command followed immediately by the RETRY gadget displays significantly different results.

CANCEL This gadget aborts the Nuke Task Selector without taking any action.

15. Nuke is capable of finding a program run, from the CLI, by using the Workbench interface Nuke procedure, provided that the name of the program run from the CLI is the same as that found beneath the icon on the Workbench.

16. If you wish to remove two programs with the same name you will have to Nuke them both successively, individually (using the CLI command line argument method).

17. If you are using the Nuke Task Selector, and wish to remove ONE of two programs with the same name, you can select either. For example, if you were running two programs with the same name, such as "Ed", and one created an error (and you knew it was the second one), then from the command line procedure you would have to remove both to get the second one. However, with the Nuke Task Selector, you can remove the second one directly, without affecting the operation of the first.

ERRATA

17. Nuke has a full complement of error messages to inform you of any problems it may encounter while running. These are displayed on the default CLI window. If you are using Nuke from the Workbench then the messages will not be available, however Nuke will beep the screen to signal you that an error has occurred with normal Nuke operation.

GOMF3.0 can be removed from the system, should the user find this desirable, by using the Nuke command. You would enter -

Nuke GOMF3.0

at the CLI prompt. Alternatively, you may Nuke GOMF3.0 from the Workbench by holding down the shift key for extended selection, click once on the Nuke icon, and then double click on the GOMF3.0 icon. You can also use to Nuke Task Selector for this purpose.

Due to the dynamic nature of the Exec task queues, you may need to try this a few times before Nuke can successfully remove GOMF3.0.

When GOMF3.0 has been successfully Nuke'd it will display a sign off message.

One likely reason a user may wish to Nuke GOMF3.0 is due to the fact that not all popular programs behave themselves, as laid out in the official Commodore/Amiga manuals. For example, LexCheck, StarGlider and Marauder II both trash the low memory Trap and Exception vector tables, early on in their initialization, upon loading. Of course (If vector-table checking has been enabled) GOMF3.0 is protecting you from this very thing! Neither of these two programs actually cause the Amiga to crash at this point and otherwise are of no problem. However, neither is therefore compatible with GOMF3.0. If you encounter a program, commercial or otherwise, that presents just this situation, then you will have no alternative but to Nuke GOMF3.0 before running the non-compliant program, or then, after you are finished with it, re-enable GOMF3.0, by entering GOMF3.0 at the prompt or using Workbench and double clicking on its icon.

GOMF3.0 facilitates the power of the Nuke command. Nuke will not work without GOMF3.0 installed into the system. If an error occurs, during a Nuke or Recall operation, which would normally go directly to a Guru Meditation Alert, you will receive a 'The Error Handler' requester. This is a special case event. The Nuke or Recall is preempted and is not carried out. GOMF3.0 has been coded to provide maximum security. This means that if an error occurs either by the program being Nuke'd or another unrelated problem, you will still be protected. After you have handled the particular error that may occur in this manner, you may again attempt to Nuke the desired program.

The Recall Command

1. The purpose of this command is to recall or activate the GOMF3.0 requester. This is usually necessary because the display was not fully cleaned up when the program was active previously. It may be that a Guru Meditation Alert preempted normal Nuke operations and that GOMF3.0 must be recalled to clear away any superfluous display elements from a previous program.

2. Recall is also a useful way to determine if GOMF3.0 is still in fully functional order should you have doubt after an abnormal circumstance occurs.

3. In the case of a Recall, the GOMF gadget does not remove programs, but only returns to normal processing.

4. Recall is a reasonably tame command to use. It may be used from either the CLI or the Workbench. If Recall does not find GOMF3.0 resident it will flash the screen in error if used from the Workbench or print an error message to the CLI if it is used that way.

5. The RECALL command is useful if it becomes necessary to reenter the error handler because you have exited it prematurely. The Error1, Error2, and Error3 programs are provided so that you may test the operation of GOMF3.0 and familiarize yourself with its operation before you attempt to beat the Guru under normal operation conditions. Error4 is a program with an 'endless loop' functional error that is unfriendly in a multi-tasking negative way. (That's computerese for saying it slows down the system.)

Chapter 7 - Examples

The Errors Drawer

1. The programs contained in this directory are provided to allow new users to become accustomed to using the GOMF3.0 program.

2. Error1 - This program is a window that displays various error gadgets, some of which the original shareware (demo) version of GOMF1.0 cannot handle. These gadgets are the Excpt Vects, MOVE SR,<ea> and Free Twice units. The first would not create an immediate error normally but would probably, in time become a serious problem. The second new gadget "MOVE SR,<ea>" causes an error only on machines 68010/20 machines. (Please see the sections IF YOU HAVE A 68010 OR 68020 PROCESSOR or Workbench Setup or ADDITIONAL GOMF STARTUP FUNCTIONS.) The third gadget causes the dreaded direct Guru Meditation Alert. There is no system requester in this case. GOMF3.0 steps in and handles this error, GOMF1.0 cannot.

3. Error2 - This program is like Error1 with its window and gadgets (except for the MOVE SR,<ea> gadget) but it opens up a custom screen and fills it with windows to practice on.

4. Error3 - This program is just the custom screen mentioned above, without the windows. After it's run there is a slight delay followed by a full Guru Alert.

5. Error4 - This is a program with no display. It does not crash the system but contains a functional error. If it is run more than once the speed degradation caused by an endless loop will be apparent. Practice Nuke'ing this program also. You may find that Nuke requires a few attempts at finding a program with this type of error.

6. Error5 - A program which will cause a complete system lockup, and cannot be bypassed (unless you have the "GOMF BUTTON" installed)

7. Error6 - This little gem enables Error5 and while trying to load Error1, will lock up and leave both drive lights on. Again, you won't recover from this error without the "GOMF BUTTON"

Questions & Answers

Q: Why doesn't GOMF3.0 have a 'HOT KEY' to call it up instead of the RECALL program?

A: We decided that the degradation in performance, created by the increase in the number of programs sifting through the input stream of the console device for special keystroke macros should be taken care of by a single shell such as CONMAN or FUNCKEY (both public domain utilities available on Fred Fish disks and elsewhere, see Appendix B), rather than by each program that requires a HOT KEY itself. It is of paramount importance that GOMF3.0 be as functional as possible, and every measure has been taken to ensure that the system's performance is maximized.

Q: How can I use GOMF3.0 to debug a program that crashes on execution?

A: Have POPCLI, or equivalent, active as a background task. When the GOMF3.0 requestor appears, pop a CLI up if none is available, run your debugger, and disassemble your code at just before the PC displayed in the 'Error Handler' requestor. You can then inspect the output of either the assembler or compiler for the source of your problem.

If the PC is pointed at ROM code, then you likely entered a random location (caused by an invalid indirect address register) or the wrong values were found in the CPU registers by the ROM code. If you can ascertain which section of code in the ROM is executing, (by figuring which library is accessed and then, using the Exec manual offsets into the jump table) you may be able to tell which of these two causes you are dealing with. This will then give you a fair idea of which of your routines is at fault.

Use a flag, which you set upon entry, to a subroutine within your program. You can view this info from within a debugger (give it a symbolic name). Then, by checking these flags, you can determine which subroutine caused the problem.

Q: Why aren't more features, bells & whistles, fancier graphics, sound, etc., etc., incorporated in GOMF3.0?

A: GOMF3.0 was written in 100% assembler and every effort was made to keep the code size small, while giving the user the best complement of features possible. We therefore left out any features which we felt could be handled by procedures, as laid out in this manual, or by readily available public domain software, such as POPCLI or CONMAN for example.

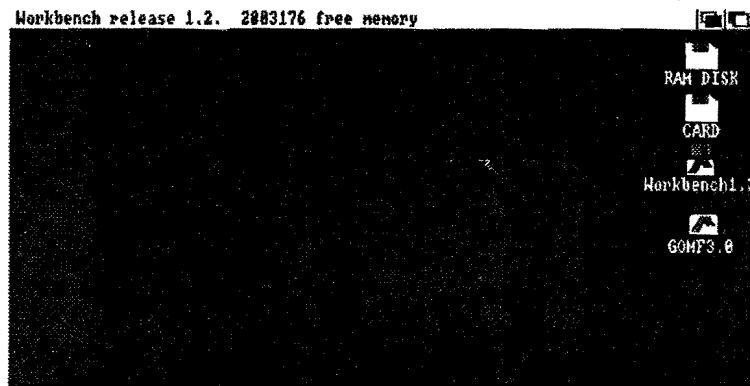
Chapter 8 - Tutorial Section

Beginner's Tutorial

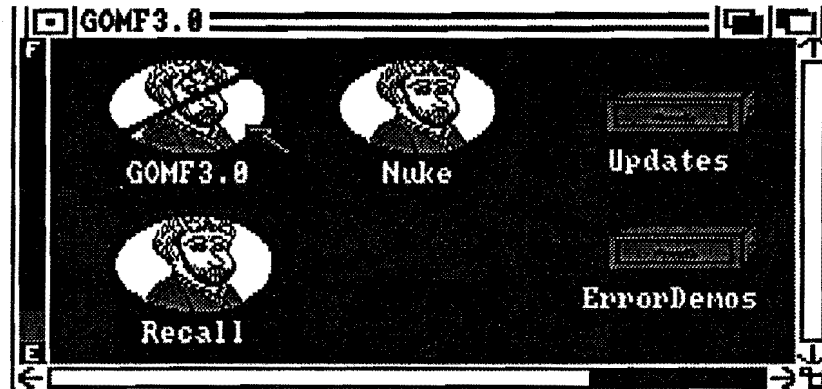
Welcome to the wonderful world of NO GURUS. In this section, we will take you step by step through error handling with GOMF. Use this book and sit down with your computer, practicing the exercises as they come along. If you find that you need information that is more technical in nature, please refer to the Advanced Tutorial Section or the appropriate section in the main body of the manual.

STARTING UP

Turn on your Amiga and place your Workbench Disk in the drive. Shortly you should see what is called the "Workbench Screen". Remove the Workbench Disk and replace it with your GOMF diskette. In a few short seconds you will see the GOMF icon on the screen. (for those who don't know, an icon is simply a graphic representation of a program or file. In this instance it looks like a little picture of a disk with the words GOMF3.0 written underneath it.)



Now we are ready to begin. Double-Click on the GOMF icon to view the directory of the disk. Shortly you will be presented with the following display.....



This is what is inside all of them:

GOMF3.0 - This is the actual program itself. The GuruBuster!

NUKE - This is a part of GOMF that will kill the kinds of errors that don't actually crash the machine, but take up memory and slow things down.

RECALL - This is also a part of GOMF which is used in circumstances where it is necessary to recall GOMF to remove more than one window or multiple errors. Don't worry about it right now. It is best to refer to the main manual to use this function.

Errors - This "Drawer" contains sample errors for you to practice

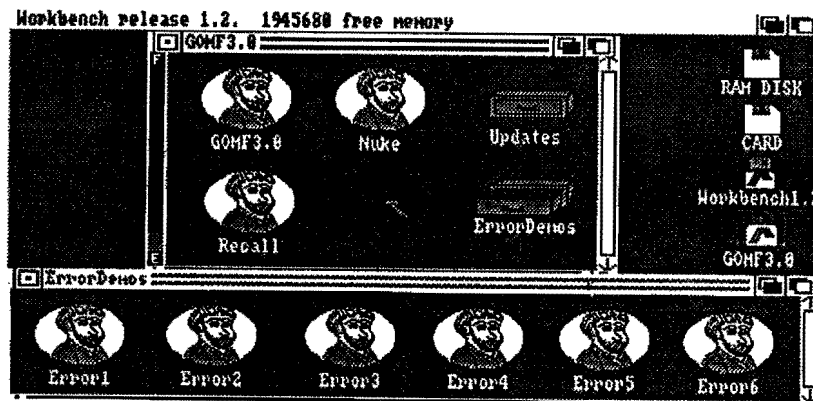
with. We will be using files from within this drawer extensively in this tutorial.

Double-Click on the GOMF icon (the one inside the window you just opened. It looks like a little picture of a guru) and you will be presented with the following display:

GOMF 3.0 'the Gurubuster' © 1988

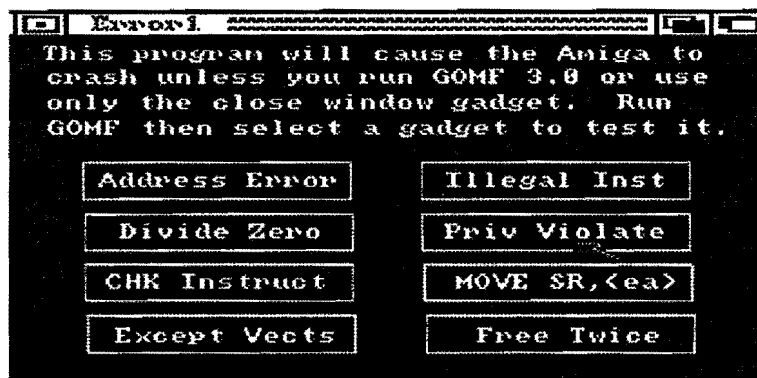
© Hypertek/Silicon Springs
This version of GOMF is not a public
domain, nor shareware, release.
Please respect this
notice of copyright.

GOMF is now resident and will trap all errors for you, automatically. Now we are ready to begin. Double-Click on the Errors drawer to reveal the following:



Practicing with Error1

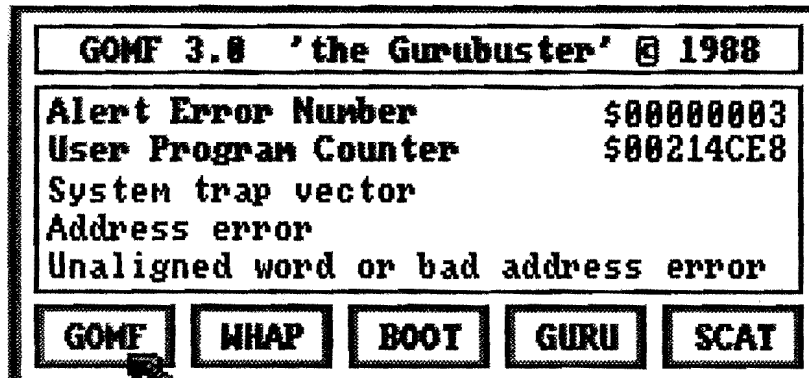
Double-Click on the Error1 icon and the following menu will appear:



The instructions at the top of the window suggest that "This program will cause the AMIGA to crash". This is absolutely true, as you will soon see.

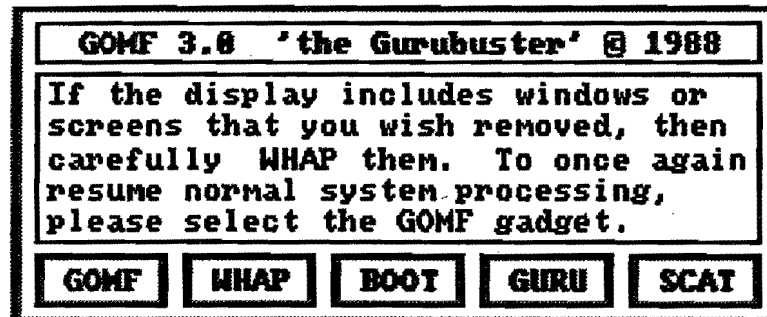
There are a great many types of errors that are possible, and several of the more common ones are shown here. We won't go into detail about exactly what happens, but suffice it to say that without GOMF, your machine will crash.

To begin, Click once on the gadget (small box) that says "Address Error". Almost instantly you will see the familiar System Request telling you to finish all disk activity and select CANCEL. So, do as you're told and click on CANCEL. The GuruBuster logo will appear and you will soon see the GOMF window telling you what happened.

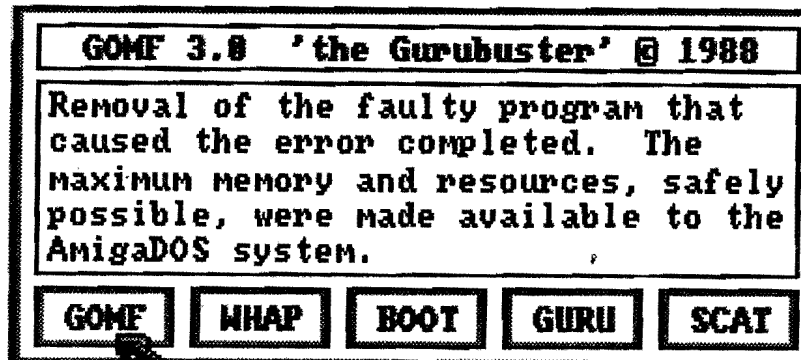


There is no need to concern yourself with what exactly happened, just be aware that an error has occurred. Underneath the 'Status Display' you will see a number of gadgets to choose from. Use them wisely, grasshopper.

This type of error is called a "Task Held Error". It has occurred in a window, namely the Error1 window. To remove the "Task" (program) that caused the error, select WHAP and then quickly click on the Error1 window. Soon the Error1 window will disappear and the GOMF window will return, giving you the following instructions:



Now, select the GOMF gadget to resume normal processing, and the following window will appear:



Everything is now back to normal. GOMF has removed the program that caused the error and your machine is still running. Notice that you did not see the famed flashing red GURU MEDITATION ALERT. Notice that you did not have to reboot your machine.

There are still three more gadgets that you have not tried yet, and they are perhaps the least dangerous.

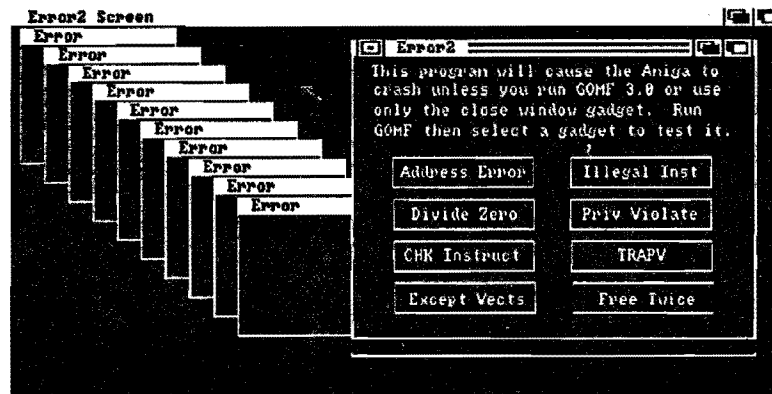
BOOT - If you select this gadget, it will reboot the machine. This has the same effect as the three-fingered Vulcan Nerve Pinch (or better known as Ctrl-leftA-rightA all at once). You should use BOOT if you suspect that other data in the machine has suffered damage due to a severe or bizarre crash. (Thankfully, this is rare!) **GURU** - If you have the suicidal urge to see the GURU and ignore the errors, click on this gadget and be prepared lose any other data that might be in the machine at the time.

SCAT - A command given to small domesticated animals, usually feline. (Oops!) Well, really it is the same thing. This gadget causes the GOMF window to move to the lower right hand side of your screen (see

illustration on next page). Clicking it again moves it back to the top left.

Practicing with Error2

Going back to the Errors drawer, click on the icon for Error2. Very shortly, you will see a new screen has opened and is slowly creating little Error windows. After a while, another window will appear with a number of errors for you to choose from, similar to the ones in Error1.



The Workbench will then cover up the Error2 windows and screen. In order to look at it, you must "drag" the Workbench screen down slightly. Click on the top bar of your screen, and while holding down the left mouse button, drag the screen down just enough to see the Error2 selections. If you still can't see both the GOMF window and the Error2 window, you may have to drag the Error2 window up its screen as well.

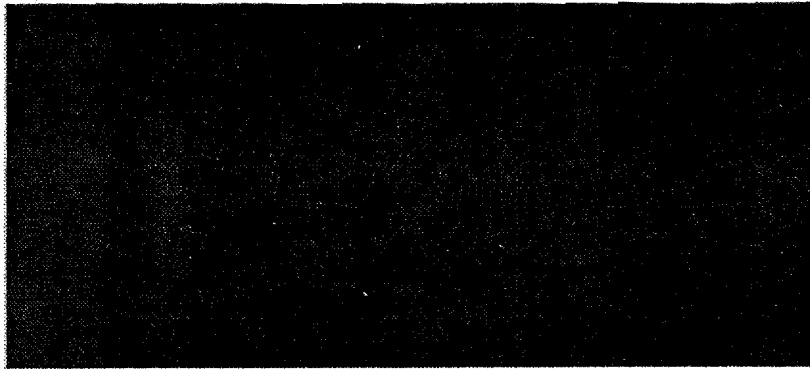
Choose the gadget that says Free Twice. Click on the WHAP gadget and then on the Error2 window. Again, as in Error1, the GOMF window will appear and ask you to select the GOMF gadget to continue. Once you have done so, the window will tell you that the maximum memory and resources have been returned to normal. You may now drag

the Workbench back up to the top of the screen.

Practicing with Error3

Back at the Errors drawer, select the Error3 icon. Almost instantly, the GOMF window will show up. Drag the Workbench window down and you will see a blank screen with the title Error3. See below for a picture of this type of error screen.

Error3 Screen



In this instance, the GOMF window will come up and tell you that an error has occurred. If you drag your Workbench screen down, you will be able to see the window again. Select GOMF. GOMF will tell you that maximum memory, etc. was safely allocated and that the system is back to normal.

The error we just handled assumed that GOMF had been loaded BEFORE the program which caused the error was loaded. When GOMF is installed before all other programs, it will automatically locate windows and screens. However, if it should occur that your program has been loaded before GOMF, you will have to tell GOMF what windows/screens to remove.

If this should happen, the message on the GOMF requester will tell you that you will have to WHAP the window before selecting the GOMF gadget. Do so, and the offending window/screen will disappear.

WHAP NOTE:

When you are instructed to WHAP a window or screen, click on the 'front-gadget' (The one at the top right-hand side) of the offending window or screen.

Up until now, we have dealt with errors that we can see, such as screens and windows. However, there are errors that can not be seen and just sit there, take up memory and slowing down the machine.

Practicing with Error4

Go back to the Errors drawer and select Error4. You will see that no errors are plainly visible, although the cursor changed briefly to its "sleeping" symbol. But, there is still an unwanted program somewhere.

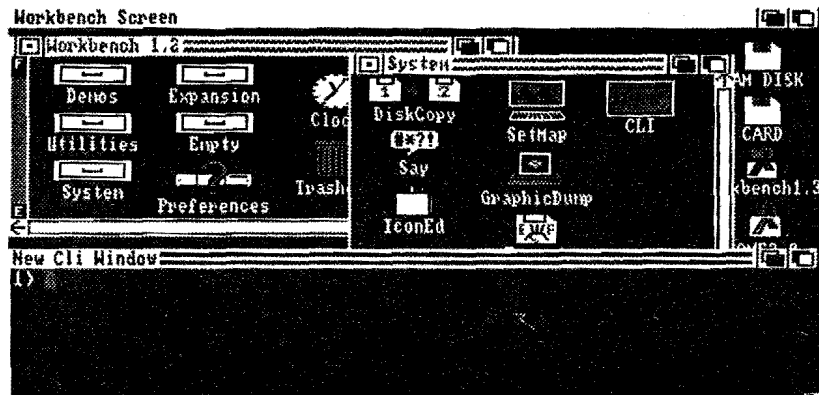
But, how do you know when this type of thing happens? A good example is if you remove a disk from one of your drives and the disk icon does not go away from the Workbench screen. This means that parts of a program are still in your computer taking up memory. This is the type of situation that Error4 is supposed to represent.

Dealing with this type of error requires that you use the CLI (Command Line Interpreter) if you want to be able to see the error. If you feel brave, try the commands below from the CLI. If you are not sure you can handle it, skip to the section entitled NUKE FROM THE WORKBENCH. Relax, it really isn't as terrible as it sounds! All it really does is to allow you to type in your commands, an alternative to using icons. It could be compared to the way the IBM compatible machines operate.

HOW TO USE THE CLI

In order to use the CLI, we must first have one. Double-Click on your Workbench disk icon. When it has finished loading, Double-Click on the System drawer. Inside you will see an icon that says CLI. Double-

Click on it, and you will see a CLI window is now active. See diagram below.



NOTES ON CORRECT CLI USAGE

The CLI assumes that whatever you are trying to load is on the same disk that the CLI is on. Depending on how many drives you have, refer to the appropriate section below for instructions. **IF YOU HAVE ONE DISK DRIVE**

Type in the following:

```
cd GOMF3.0:
```

The Amiga will prompt you with a “System Requester” and ask you to ‘place volume GOMF3.0 in any drive’. Take out your Workbench and replace it with your GOMF disk. You don’t have to click on the retry gadget. The computer will do it by itself as soon as it realizes that the disk has been replaced.

IF YOU HAVE TWO DISK DRIVES

Assuming your Workbench is in drive 0 and GOMF is in drive 1, type in the following:

```
cd df1:
```

ALL TOGETHER NOW

There. Now that was relatively painless, wasn’t it! The reason we had to use the preceding commands is because the CLI always assumes that whatever you are looking for is on the same disk as the CLI. This wasn’t the case, so by entering a very few commands, we are now able to continue.

Now, type in the following command, and hit enter:

```
nuke !
```

You should now see a listing, similar to what is shown on this page.

```
AnigaShell
Task Ready Queue
NEWCON
GOMF3.0
Task Waiting Queue
Background CLI
viend.device
SYS: System/CLI
RAM
Workbench
trackdisk.device
trackdisk.device
CARD
File System
File System
DF1
trackdisk.device
input.device
Press return for more.
```

The various names you see are really not important, and they may very well be different on your machine, so don't worry. All that matters is that you can see Error4 in the Task Ready Queue. Type in the following and hit enter:

nuke Error4

Make sure that you enter it exactly as shown, otherwise it won't work. It matters a lot whether you use upper or lower case letters.

If everything worked, you should see the GOMF window (except

the GOMF gadget now says NUKE).

```
GOMF 3.0 'the Gurubuster' © 1988
To completely remove the program you
have selected, use WHAP to remove
the display. If applicable, you may
use automatic display removal. Then
exit by selecting the NUKE gadget.
NUKE WHAP BOOT GURU SCAT
trackdisk.device
input.device
AnisabOS Task Array
nuke
c:SetAlert
Nuke arguments are case sensitive.
1> nuke Error4
```

Since there are no display elements (windows or screens) select the NUKE gadget, as requested. Shortly, the CLI will respond with:

Nuke Completed Task Removal Without Error

When you type 'nuke !' again, the Error4 will have disappeared from the Task Ready queue. Pretty simple, huh.

If you happen to try this on your own programs, be aware that whatever the name of the program is, it must be typed EXACTLY as it appears in the task queue.

For example, if your display is like this:

CON
GOMF3.0
FILENAME.ext

You **MUST** type it in exactly, with the right type of case (upper or lower). The correct usage would look like this:

nuke FILENAME.ext

If the filename was FILE NAME.ext (with a space between FILE and NAME), place it in quotation marks:

nuke "FILE NAME.ext"

Any time that a name has spaces or other punctuation, place the filename in quotations and it will work.

It really is very straight-forward. There are a million possibilities, and since we are limited by space, we suggest that you gain more information through user's groups, books, magazines, or your friendly dealer. If you come across something that you just can't handle, refer to the section in the GOMF manual titled PRE-CALL checklist before you call us for help. We are happy to oblige, but humour us by reading the manual first.

GETTING OUT OF THE CLI

Just type ENDCLI and hit enter.

NUKE FROM THE Workbench

Well, hopefully you tried to use NUKE from the CLI. We gave you the most difficult way first. There are two ways to run NUKE from the Workbench, but you had better be careful! You can very easily cause more

harm than good.

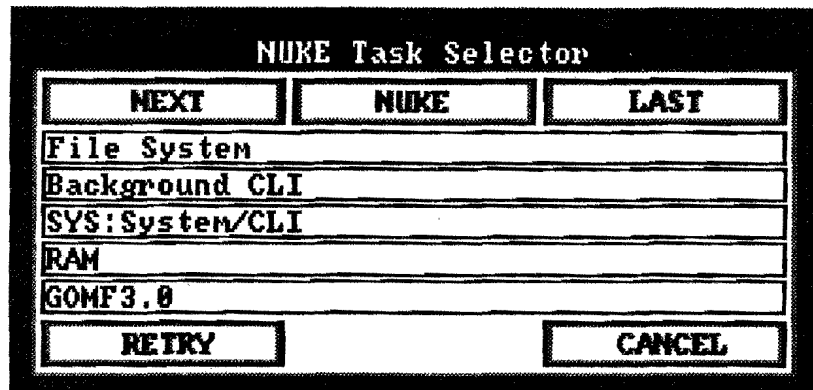
NUKE USING EXTENDED SELECTION

If you look back to when you originally booted up your machine, you will see that as well as the GOMF3.0 icons, there is an icon called **NUKE**. This is the same thing as the **NUKE** you may have just tried from the CLI. To use it, we use a feature of the Amiga called extended selection. It works like this:

1. Make SURE that GOMF has been loaded, otherwise.....[say bye bye]
2. Click ONCE on the **NUKE** icon (you will then see a black diagonal line, crossing out the face of the little guru inside)
3. While holding down either 'shift' key on the keyboard, Double-Click on the program you wish to remove.
4. If your particular circumstance causes an error window to appear, simply handle it as any other error. Just be CAREFUL !!!

NUKE USING THE NUKE REQUESTER

There is an even easier way of using **NUKE** from the workbench. This method is used by simply double-clicking on the **NUKE** icon on your GOMF disk.



The gadgets operate as follows:

TASK

FIELDS These fields contain the names of the tasks or processes currently in your system. To select a task you want to NUKE, simply click on the gadget containing that name. This will cause that particular gadget to remain selected. When a task field gadget is selected in this manner, it will be removed by selecting the NUKE gadget. To deselect a task, click on it once again, or select any other gadgets.

NEXT This will display the next screen of five tasks. Occasionally, a blank field may appear. You cannot select a blank field.

LAST This gadget will display the previous five tasks.

NUKE Selecting NUKE causes the currently selected task or

and subsequently Nuke'd at a time. If no task is selected, nothing will be Nuke'd.

RETRY The purpose of RETRY is to reload the task field gadgets. It should be noted that there are a few transient states which tasks can occupy which mean that the task is temporarily in none of the system lists. Also, due to task switching and scheduling, a program may simply be between lists, in the process of being added to one while being removed from another. You may well find that one use of the Nuke command followed immediately by the RETRY gadget displays significantly different results.

CANCEL This gadget aborts the Nuke Task Selector without taking any action.

NUKE NOTES

Nuke is capable of finding a programrun, from the CLI, by using the Workbench interface Nuke procedure, provided that the name of the program run from the CLI is the same as that found beneath the icon on the Workbench.

If you wish to remove two programs with the same name you will have to Nuke them both successively, individually (using the CLI command line argument method).

If you are using the Nuke Task Selector, and wish to remove ONE of two programs with the same name, you can select either. For example, if you were running two programs with the same name, such as "Ed", and one created an error (and you knew it was the second one), then from the command line procedure you would have to remove both to get the second one. However, with the Nuke Task Selector, you can remove the second one directly, without affecting the operation of the first.

Nuke has a full complement of error messages to inform you of

any problems it may encounter while running. These are displayed on the default CLI window. If you are using Nuke from the Workbench then the messages will not be available, however Nuke will beep the screen to signal you that an error has occurred with normal Nuke operation. GOMF3.0 can be removed from the system, should the user find this desirable, by using the Nuke command. You would enter -

Nuke GOMF3.0

at the CLI prompt. Alternatively, you may Nuke GOMF3.0 from the Workbench by holding down the shift key for extended selection, click once on the Nuke icon, and then double click on the GOMF3.0 icon. You can also use the Nuke Task Selector for this purpose.

You may need to try this a few times before Nuke can successfully remove GOMF3.0. When GOMF3.0 has been successfully Nuke'd it will display a sign off message.

One likely reason a user may wish to Nuke GOMF3.0 is due to the fact that not all popular programs behave themselves, as laid out in the official Commodore/Amiga manuals. For example, LexCheck, StarGlider and Marauder II both trash the low memory Trap and Exception vector tables, early on in their initialization, upon loading. Of course (If vector-table checking has been enabled) GOMF3.0 is protecting you from this very thing! Neither of these two programs actually cause the Amiga to crash at this point and otherwise are of no problem. However, neither is therefore compatible with GOMF3.0. If you encounter a program, commercial or otherwise, that presents just this situation, then you will have no alternative but to Nuke GOMF3.0 before running the non-compliant program, or then, after you are finished with it, re-enable GOMF3.0, by entering GOMF3.0 at the prompt or using Workbench and double clicking on its icon.

GOMF3.0 facilitates the power of the Nuke command. Nuke will not work without GOMF3.0 installed into the system. If an error occurs, during a Nuke or Recall operation, which would normally go directly to a Guru Meditation Alert, you will receive a 'The Error Handler' requester. This is a special case event. The Nuke or Recall is preempted and is not carried out. GOMF3.0 has been coded to provide maximum security.

This means that if an error occurs either by the program being Nuke d or another unrelated problem, you will still be protected. After you have handled the particular error that may occur in this manner, you may again attempt to Nuke the desired program.

The Recall Command

The purpose of this command is to recall or reactivate the GOMF3.0 requester. This is usually necessary because the display was not fully cleaned up when the program was active previously. It may be that a Guru Meditation Alert preempted normal Nuke operations and that GOMF3.0 must be recalled to clear away any superfluous display elements from a previous program.

Recall is also a useful way to determine if GOMF3.0 is still in fully functional order should you have doubt after an abnormal circumstance occurs.

In the case of a Recall, the GOMF gadget does not remove programs, but only returns to normal processing.

Recall is a reasonably tame command to use. It may be used from either the CLI or the Workbench. If Recall does not find GOMF3.0 resident it will flash the screen in error if used from the Workbench or print an error message to the CLI if it is used that way.

The RECALL command is useful if it becomes necessary to reenter the error handler because you have exited it prematurely. The Error1, Error2, and Error3 programs are provided so that you may test the operation of GOMF3.0 and familiarize yourself with its operation before you attempt to beat the Guru under normal operation conditions. Error4 is a program with an 'endless loop' functional error that is unfriendly in a multi-tasking negative way. (That's computerese for saying it slows down the system.)

Practicing with the "GOMF BUTTON"

If the package you purchased contained the GOMF BUTTON

hardware device, follow the instructions in Appendix C for the installation. If you have any doubt whatsoever as to your ability to install the device, PLEASE take your computer, the GOMF BUTTON and this manual to any qualified technician who repairs AMIGA computers and he will do it for you, likely for only a very small fee. You may find that the best place to begin is at your local dealer.

Using Error5

After opening the Errors drawer, select Error5. A small box will appear, explaining that your Amiga is about to lock up and die. Shortly, you will notice that moving the mouse does not move the cursor. The computer is now dead. Or is it? Press your GOMF BUTTON. Immediately, your cursor will have sprung to life and your computer has returned from the dead! Amazing, ain't it?

You have just survived the most sinister and deadly of all errors, affectionately known as a 'catastrophic system failure'. If this were a real program, you would now attempt to save any data you may have been working on. After you have done so, it would be a fairly good idea to reboot your machine-just to play it safe.

Using Error6

This selection is similar to Error5, except that the disk drive lights will remain on while a freeze occurs.

Advanced Tutorial Section

MULTIPLE ERROR HANDLING GUIDELINES

GOMF3.0 has been enhanced to handle multiple errors occurring during what has become known as 'normal' operation of the program. Due to the complex nature of the system during error handling it is a good possibility that errors may occur while coping with the initial source of a particular problem. GOMF3.0 has been coded to support limited re-entrancy. This means that it will attempt to survive multiple occurrences of errors. It will handle any errors received during it's active, 'requester present' state, in a preemptive manner. This increases its reliability quite a bit. However, this also increases the amount of user expertise required to support this function.

The majority of these types of occurrences are related to the systems' handling, either of display objects that are no longer attached to ports properly, or Exec tasks that no longer behave themselves due to their current state of affairs. This is not to say, of course, that GOMF3.0 is not also responsible for this on occasion. Such is the nature of survival in this very likeable computer.

Should an additional error occur during a normal Nuke, Recall, GOMF, WHAP operation, or simply a system error occurred due to one of the aforementioned reasons, there is a good chance that GOMF3.0 may allow the user to yet again keep the system afloat. Kind of exciting, don't you think?

It is suggested that you follow through this section of the manual mouse in hand, as it were. This will give you first hand experience in beating the Guru when he's really down and dirty. If you have a working knowledge of the CLI, then it would be wise to have one around for these manoeuvres. Although you can accomplish this strictly from the Workbench, you will be able to 'see' the task and process structures as they are worked over with a CLI window with which to interact.

MULTIPLE ALERT ERRORS

First and foremost you must have GOMF3.0 installed into the system. See the section "Using GOMF3.0" in the main manual if necessary. CLI users are recommended to LoadWB at this point. Insert the GOMF3.0 diskette into a drive and double click on the disk. Next, open the Errors drawer. Double click on the Error1 icon. Once the program window has opened, move it down to the lower left hand corner of the display. Double click on the Error1 icon again. Move this program's window to the lower right hand side of the display. You now have two identical programs in memory.

The purpose of this is to create multiple errors for you to practice upon, in the comfort of your own computer.

Use the righthand Error1 window to cause the first error. Select the Free Twice gadget. This causes a direct Guru error. The GOMF3.0 requester appears. Click on its SCAT gadget. This will have moved the requester to the lower righthand corner of the display, overlaying the offending Error1 program window. Now for the evil deed! On the lefthand Error1 window select the Free Twice gadget. This will create a second direct Guru.

You will notice that the second iteration of the Error1 program on the left hand side of your screen has vanished. What has happened is that GOMF3.0 has removed the second Error1 program to the best of its ability. If there any display elements left, they may now be removed with the WHAP gadget. You may now proceed with the normal removal of the first Error1 program on the right-hand side of your screen.

ERRORS ENCOUNTERED DURING NUKE OR RECALL

To test this, you must follow the initial instructions for the multiple

error handling portion of this tutorial. You must have GOMF3.0 installed into the system. Have a CLI window available, if desired. Also, run the Error1 program twice, and arrange the windows, as before.

This time though, click the GOMF3.0, Errors directory window to the front, over top of the righthand Error1 window. Follow the instructions under the section Using Nuke to attempt a Nuke. This means you will either type Nuke Error1 at the CLI prompt or, using extended selection, hold down the shift key, click on the Nuke icon, double click on the Error1 icon and then release the shift key. The GOMF3.0 requester will appear, it will have the GOMF gadget transformed into the NUKE gadget. Click on its' SCAT gadget to move the requester to the lower righthand side of the display.

Now simulate an error during the Nuke operation. To do this select one of the error gadgets of the lefthand Error1 program's window. If this was a Task Held type error select the CANCEL gadget on the system requester.

You will notice that the second iteration of the Error1 program on the left hand side of your screen has vanished. What has happened is that GOMF3.0 has removed the second Error1 program to the best of its ability. If there any display elements left, they may now be removed with the WHAP gadget. You may now proceed with the normal removal of the first Error1 program on the right-hand side of your screen. If all went well you have just survived a catastrophic error while attempting to remove a program that itself contained some sort to functional error.

HAVE AT IT!

Working with the system under these types of conditions is analogous to what it must be like to work in an emergency ward of a hospital on Saturday night. No sooner do you stabilize the situation when some other problem blows in the door.

If you are unsuccessful in your first attempts at these practice sessions, please try again. Remember, sometimes you may be able to

reorganize the screen display during these occurrences, but for maximum reliability don't chance it. It goes without saying that under 'real conditions', when you are using your own regular programs, the handling of such multiple errors is more tricky because you may not know which program in the multi-tasking environment has caused the error.

Keep in mind that any errors that cause the normal system Task Held requester, should be left until the last during any mop up operations. Any programs in this state are neutralized by the system and will be of no immediate danger. Simply finish normal procedures, ignoring the requester, while being careful not to WHAP it. Then, deal with the cause of the system requester.

Once you have coped with errors preempting error handling a few times, using these procedures and cautions, you will build your expertise and confidence in restoring the system to normal.

CHAPTER 9 - ADVANCED APPLICATIONS

Gomf1.0.obj is an object code file generated by an assembler. For a programmer to use it the code must be linked to the object output of an assembler or compiler. The ALINK linker directives FROM or ROOT are used to accomplish this. For example...

```
ALINK MyProg,GOMF1.0.obj to ProgName
```

GOMF1.0.obj provides the same protection as GOMF1.0, however the linked module allows the programmer to design his or her own method of handling these error returns.

Setting up your source code for use with Gomf1.0.obj is relatively easy. Your code must make references to the following external labels, _GOMF and _GOMFEnding. You may also wish to utilize other information with external reference labels of _WHAP, _ProgramCounter, _GeneralErr, _LibraryErr, _SpecificErr. Gomf1.0.obj requires your source define it's own external label, _ErrorHandler, which is coded to receive the return to normal processing, after the error has been neutralized. You will have to write up a custom routine called _ErrorHandler that will receive the program flow after an error has been detected elsewhere in the program. This may be simply closing screens, windows and libraries before exiting, or you may wish to analyze the situation more completely and continue execution of your program code at another point. The potential of this feature is not small.

Early on in your program do a simple call of _GOMF such as JSR _GOMF or GOMF(). This will activate the features of the linked module. Before your program cleans up and ends you must call _GOMFEnding in a like fashion so that _GOMF can release it's memory usage etc.

To recap, _GOMF is the initialization entry point. _GOMFEnding is the termination entry point. The other labels available are a structure

laid out as follows...

STRUCTURE	GOMF1.0,0	
ULONG	_ProgramCounter	value of the program counter after the error
APTR	_LibraryErr	pointer to null terminated string descriptor
APTR	_GeneralErr	pointer to null terminated string descriptor
APTR	_SpecificErr	pointer to null terminated string descriptor
UBYTE	_WHAP	boolean TRUE or FALSE of WHAP gadget selection

See the source code example of the use of this structure in Err4.asm
for more clarification.

Chapter 10 - TroubleShooting

At some point, you will probably find a way to crash your Amiga even with GOMF3.0 installed. (in spite of your commendable efforts at taking every precaution!) The following info is presented as a means of partial consolation.

1. There is an officially undocumented Workbench feature that you may wish to experiment with. Try the following:

Reboot your Amiga (CTRL-Amiga-Amiga)

Press CTRL-D to BREAK to CLI

Type: LOADWB -DEBUG

ENDCLI (or resize the CLI window to reveal the Workbench.)

Now, from the Workbench screen, press the right mouse button to examine the menus, and marvel, if you will, at the secret DEBUG menu. Although documentation is scarce concerning this feature, it seems that if you just happen to have an external terminal (such as another computer running terminal software) connected via a null modem to the Serial port, you can perform disassemblies, etc. with this menu.

In a similar vein, if you select the GURU option from GOMF's Requester, (or aren't running GOMF and meet the GURU), you will crash and see the evil red "press left mouse button to continue" Alert message. Well, did ya ever wonder what would happen if you DISOBEYED the GURU and pressed the RIGHTBUTTON instead? A terminal connected, as before, will dump reasonably useful diagnostic messages out through the serial port.

O.K., so maybe those weren't that useful, after all. The following is a checklist of things to look for if you find your machine crashing a lot.

The Pre-Panic Checklist

1) Could it be that the software you are attempting to run is designed for a different version of Kickstart than the one you are using? Remember

that the Amiga has enjoyed/suffered at least three major revisions of the Kickstart code, and a fair number of older programs do not work on the newer kickstart versions.

For example, Kickstart 1.0, which was shipped with the original Amiga 1000's was rather rife with bugs, and lacked numerous features of today's Kickstart 1.2. Any software (including Workbench 1.2) which attempts to use one of these features not present in the old Kickstart will definitely CRASH.

The previous statements also apply (sans a few bugs) to Kickstart 1.1, which ALL Amiga 1000 owners received with their machines. There are many examples of programs that are Kickstart version dependant, such as TRANSFORMER, LEADER BOARD GOLF and MUSICRAFT, which do NOT work on Amiga 500/2000's or A1000's running Kickstart 1.2. Most of the earlier Electronic Arts programs also suffer from incompatibility with the new Kickstart.

The solution is simple if you have an A1000. Simply reboot with the appropriate version of KICKSTART, and the programs should function as expected.

However, if you have an A500 or A2000, your only options are as follows:

- Get an updated version of any software that is not compatible with Kickstart 1.2.
- Install a ROM (downgrade?) kit that would allow you to switch between 1.2 and the older Kickstart(s).

As a general rule, Kickstart and Workbench versions are 'downwardly compatible', meaning that the Kickstart should be at least as new as the Workbench disk you want to run with it. Therefore, a Kickstart 1.2 will support a Workbench 1.1, but NOT vice-versa.

2. Is a crash-inducing bug reproducible on another similarly-configured Amiga? If NOT, that definitely points the finger of suspicion at your hardware.(uh-oh!) Have your machine examined by an authorized

Amiga service technician.

3. **ADD MEMORY!** Amiga owners the world over rejoice with their multi-Megabyte machines. Crashes diminish substantially, system performance improves dramatically, and GEE WHIZ, you can sure multitask a lot of stuff.

4. It seems to us that crashes seem to occur most often when we are rapidly or erratically moving windows, pulling down menus, etc., WHILE the machine is trying to do something else (like load a file). The moral is: be patient and gentle with AMI, and she'll be kind.

5. Remember also, that good software is rare and precious. There are many programs that have BUGS that semi-randomly crawl out and bite you. GOMF is your best line of defence, but nothing can replace the following WISE WORDS:

**SAVE YOUR WORK...FREQUENTLY
IF IT'S IMPORTANT, BACK IT UP!**

6. If you are still having difficulties, and have exhausted these possibilities, call our technical support number. See the section entitled **CUSTOMER SUPPORT**.

Please **NOTE**: You **MUST** have sent in your GOMF3.0 registration card to be eligible for this service. If your question is of a highly technical nature, please submit it in writing. We will respond promptly to all letters from registered owners.

Credits

GOMF1.0, Gomf1.0.obj, GOMF3.0, Err1, Err2, Err3, Err4, Err5, Err6, Hey!, NUKE and RECALL were written entirely in assembly language by Christian Johnsen. The documentation was written by Graeme Bennett Christian Johnsen, and Dean Spyropoulos.

Tools

The following is a list of tools used in the creation of this product, of which we are very proud:

Cygnus Ed Pro
Power Windows
Metacomco Assembler
CAPE Assembler
Metascope
TxEd
Grabbit
Deluxe Paint
Professional Page
(All are trademarks of their respective companies)

Beta Testers

The following Wonderful People were of invaluable assistance with the production of GOMF. Hats off to:

Larry Rosenman Sterling Heights, MI	Larry Phillips Vancouver, BC
Eddie Churchill Dallas, TX	Scott Bussi Vancouver, BC
Tom Smythe Seattle, WA	James Thompson Union City CT
Richard Lambert Clearbrook, BC	

GOMF is a trademark of Hypertek/Silicon Springs Development Corp.

Amiga is a trademark of Commodore-Amiga, Inc.

Other products mentioned are trademarks of their respective companies.

We trust this application is of use and value to you, and welcome any communication via mail at the address below.

Hypertek/Silicon Springs
812 Surrey Street
New Westminster, B.C. Canada
V3L 4W2

Phone (604)-524-1125

Fax (604)-941-9358

Appendix A - Guru Meditations

Format of Guru Meditation Messages:

Subsystem ID Code	General Error	Specific Error	Address of Task
00	00	0000	00000000

Subsystem ID codes

CPU

CPU Trap	00
----------	----

Libraries

Exec	01
Graphics	02
Layers	03
Intuition	04
Math	05
Clist	06
DOS	07
RAM	08
Icon	09
Expansion	0A

Devices

Audio	10
Console	11
Gameport	12
Keyboard	13
Trackdisk	14
Timer	15

Resources

CIA	20
Disk	21
Misc	22

Other

Bootstrap	30
Workbench	31
Diskcopy	32

NOTE: If the first digit of the subsystem ID is greater than 3, there is no way to recover from the error. In these cases, subtract 8 from the first digit to get the subsystem ID number.

General Error Codes

-00 if not applicable

Insufficient memory	01
MakeLibrary error	02
OpenLibrary error	03
OpenDevice error	04
OpenResource error	05
I/O error	06
No signal	07

Specific Alert Codes

Exec Library

81000001	68000 exception vector checksum
81000002	ExecBase checksum
81000003	library checksum error
81000004	no memory to make library
81000005	corrupted memory list

81000005 corrupted memory list
81000006 no memory for interrupt servers
81000007 InitAPtr
81000008 semaphore corrupt
81000009 free twice
8100000A bogus exception

Graphics Library

82010001 no memory for copper display list
82010002 no memory for copper instr. list
82000003 copper list overload
82000004 copper intermediate list overload
82010005 no memory for copper list head
82010006 long frame, no memory
82010007 short frame, no memory
82010008 no memory for flood fill
82010009 text, no memory for TmpRas
8201000A no memory for BltBitMap
8201000B region memory
82010030 MakeVPort
82011234 GfxNoLCM

Layers Library

83010001 LayersNoMem

Intuition Library

84000001 Unknown gadget type
04000001 Recoverable version of prev. msg
84010002 No memory to create port
84010003 Item plane alloc, no memory
84010004 Sub alloc, no memory
84010005 Plane alloc, no memory
84000006 Item box top less than RelZero
84010007 No memory to open screen
84010008 Open screen, raster alloc, no mem

8401000A Add SW gadgets, no memory
8401000B No memory to open window
8400000C Bad State Return entering Intuition
8400000D Bad message received by IDCMP
8400000E Weird Msg causing incomprehension
8400000F Couldn't open the console device

DOS Library

07010001 No memory at startup
07000002 EndTask didn't
07000003 Qpkt failure
07000004 Unexpected packet received
07000005 Freevec failed
07000006 Disk block sequence error
07000007 Bitmap corrupt
07000008 Key already free
07000009 Invalid checksum
0700000A Disk Error
0700000B Key out of range
0700000C Bad overlay

RAM Library

08000001 Bad segment list

Expansion Library

0A000001 Bad Expansion Free

TrackDisk Device

14000001 Calibrate: seek error
14000002 Delay: error on timer wait

Timer Device

15000001 Bad request
15000002 Bad supply

Disk Resource

21000001 Get unit: already has disk
21000002 Interrupt: no active unit

BootStrap

30000001 Boot code returned an error

*CPU Traps (exceptions) are defined by the 68000 cpu and not by the Amiga system software. There are 256 vectors for CPU exceptions; the lower 64 are hardware-defined, the upper 192 are user-definable interrupt vectors. Specific CPU traps you might see in a Guru are:

00000002	Bus error
00000003	Address error
00000004	Illegal instruction
00000005	Divide-by-zero
00000006	CHK instruction
00000007	TRAPV instruction
00000008	Privilege violation
00000009	Trace
0000000A	Op Code 1010
0000000B	Op Code 1111
0000000C	Motorola Reserved Vector
0000000D	Coprocessor Protocol Violation
0000000E	Processor Format Error
0000000F	Uninitialized Interrupt
00000010-00000017	Motorola Reserved Vectors
00000018	Spurious Interrupt
00000019	Autovector Level 1
0000001A	Autovector Level 2
0000001B	Autovector Level 3
0000001C	Autovector Level 4
0000001D	Autovector Level 5
0000001E	Autovector Level 6
0000001F	Autovector Level 7
00000020-0000002F	Processor Trap

00000020-0000002F	Processor Trap
	Number #0-#15
00000030	FPCP Branch or set on unordered condition
00000031	FPCP inexact result
00000032	FPCP Divide by Zero
00000033	FPCP Underflow
00000034	FPCP Operand Error
00000035	FPCP Overflow
00000036	FPCP Signaling NAN
00000037	Motorola Reserved Vector
00000038	PMMU Configuration
00000039	PMMU Illegal Operation
0000003A	PMMU Access Level
0000003B-0000003F	Motorola Reserved Vectors
00000040-000000FF	User Defined Vectors

NOTES:

FPCP - Floating Point CoProcessor

PMMU - Paged Memory Management Unit

Appendix B - Public Domain Software Sources

PUBLIC DOMAIN SOFTWARE

This software is collected from user groups and electronic bulletin boards around the nation. To the best of our knowledge, the materials in these libraries are freely distributable. This means they were either publicly posted and placed in the Public Domain by their Author, or they have restrictions published in their files denoting the conditions of their distribution.

AMICUS: over 22 disks available.

Highlights:

AMICUS 1 contains useful information on using Amiga Debugging tools WACK and RomWACK, CLI info, etc.

AMICUS 8 contains a program called GuruMed, which contains a disk file of information similar to that contained in Appendix A of this manual.

AMICUS 9 contains tips on how to install a 68010 CPU, which GOMF3.0 is compatible with. Note that GOMF3.0 eliminates the need for DeciGEL (also on this disk).

FRED FISH: over 130 disks available.

Highlights:

FFDisk 2 contains DBug, a debugging utility. Newer versions on disks 41 and 102.

FFDisk 18 contains docs on installing a 68010 microprocessor.

FFDisk 27 contains Disassem, a simple 68000 disassembler.

FFDisk 35 contains PopCLI, which will, when used with GOMF, allow the user to open a CLI after a normally-fatal error for debugging, etc. Highly recommended for the intrepid GOMFer. New version on disk 40.

FFDisk 54 contains MemWatch, which, like one of GOMF3.0's functions, watches for programs that trash low memory and attempts to repair the damage. New version on disk 87.

FFDisk 58 contains the ASDG recoverable RAMDISK. Another must-have!

FFDisk 69 features ConMan, a CLI/console enhancement that adds editing and history to any application that uses CON: New versions on Disk 81 and 100.

FFDisk 79 contains KILL, which (like GOMF's NUKE command) removes a task and its resources.

FFDisk 89 features FuncKey, a shareware function key editor.

FFDisk 95 GOMF1.0! An instant classic. Also contains Journal, which records mouse and keyboard events.

APPENDIX C - HARDWARE INSTALLATION

GOMF BUTTON HARDWARE INSTALLATION

Please read the instructions! That's why they're here!

Read through the instructions for your computer model a few times to familiarize yourself with the procedure before picking up any tools. Use a pencil to check off the steps that you have completed as you go, that way if you get interrupted you'll know where to continue from.

If you don't feel confident about taking your computer apart, then please have your dealer or local computer repair center do the installation for you. It only takes 10 to 20 minutes to do the job right the first time.

TORX screwdriver #T-10 (AMIGA 500 only)

PHILLIPS screwdriver (AMIGA 1000/2000)

I.C. Extractor for 48 pin PAULA I.C., or small slotted type screwdriver

Drill and 3/16 inch drill bit

Needle nose pliers

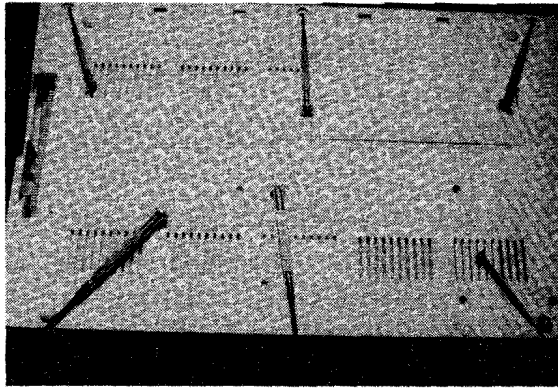
Anti-static foam (part of packaging)

AMIGA 500 INSTALLATION:

A) Unplug the computer and remove the power cord and any other cables, cords and accessories including the 512k memory expansion, if installed.

B) Refer to picture 1 to locate the 6 screws that hold the computer case together. You have to turn the computer over to find them. Using a TORX T10 screwdriver, remove the six screws indicated. You should

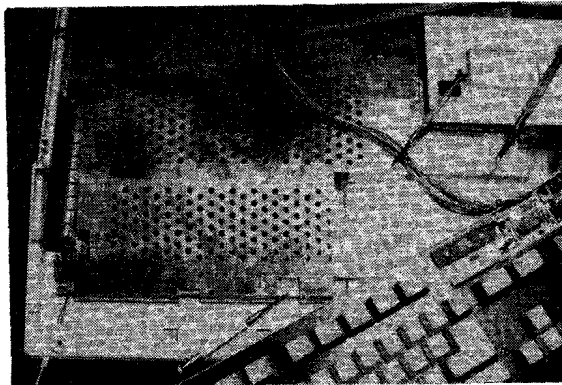
also remove the expansion port cover at this time.



Picture 1

C) Turn the computer right side up. There are two plastic catches on each side of the computer that hold the case top on. By gently wiggling the top of the case, it will eventually come off.

D) Refer to picture 2. To move the keyboard out of the way, lift up and slide it towards the back of the computer to disengage the plastic retainers that locate the bottom of the keyboard in the case.



Picture 2

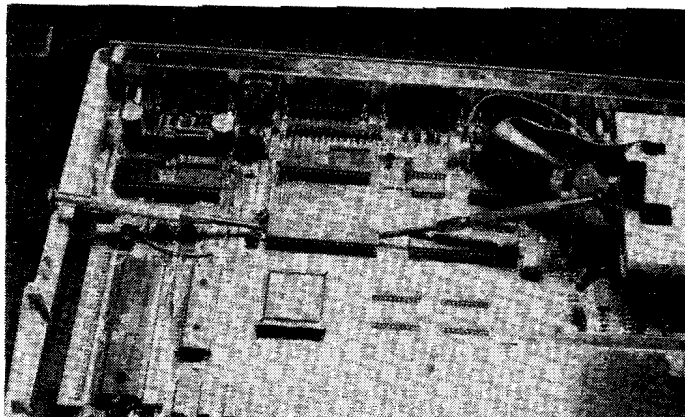
E) Locate the bundle of wires coming from the keyboard. Make a note of what color wire is on the left and right of the connector just inside the shield. Now pull gently on the wire bundle and the connector will come out.

F) There are 4 screws and 4 metal tabs that hold the top of the shield on. Bend the 4 metal tabs up by lifting them with a knife or screwdriver and straightening them with the pliers. Remove the 2 screws from the front of the shield and put them aside (separate from the case screws). The last 2 screws are located near the expansion connector. When removing them, note how the expansion connector shield is retained.

G) Lift the shield up and wiggle it off the metal tabs, then put it aside.

H) Locate PAULA. The number on PAULA is 8364R?. The number after the R is the revision number, which varies; the other number below 8364R? is the production date. NOTE THE LOCATION OF THE NOTCH ON THE END - IT MUST FACE THE SAME WAY WHEN YOU ARE FINISHED.

I) Refer to picture 3. Insert screwdriver (or use I. C. extractor) between I. C. and socket and twist or pry the Paula chip up evenly from both ends as indicated.



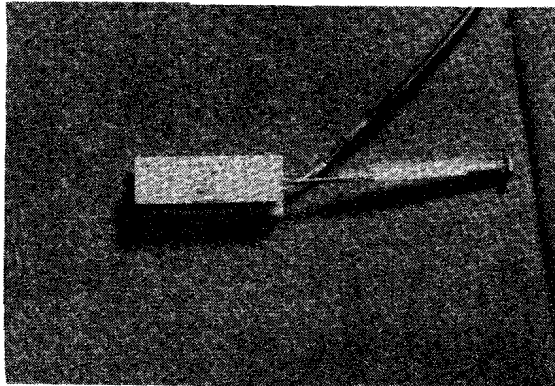
Picture 3

J) BEFORE PICKING UP THE PAULA CHIP, neutralize any static charge you might have built up by touching a cold water pipe, faucet or something known to be grounded.

K) Remove PAULA, straighten any bent leads, and place it on the foam.

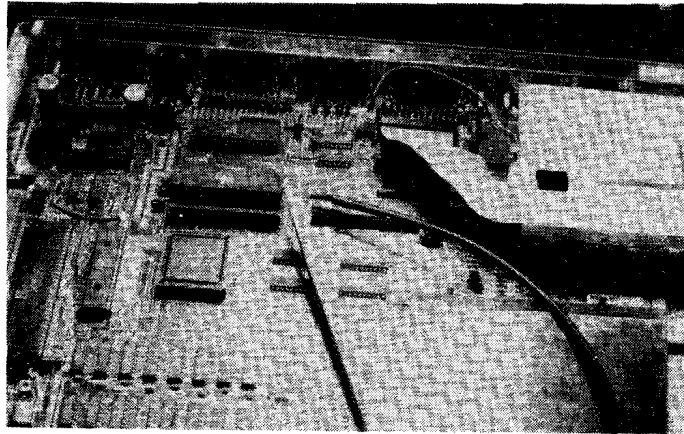
L) Locate the GOMF circuit, remove it from it's foam pad, and CAREFULLY straighten any bent pins. Check the wires connecting the button to the GOMF circuit to make sure they are not broken. If they are broken take it to a qualified electronics technician to have them re-installed. The wires go into the two large holes at the end of the circuit board. It doesn't matter which wire goes in which hole.

M) Put the GOMF circuit back in it's foam and place PAULA on GOMF so that the notch on PAULA is above the wires. Refer to picture 4. If all pins appear to line up with all the holes in the socket, firmly and evenly push PAULA straight into the GOMF socket.



Picture 4

N) Refer to picture 5. Make sure the pins on GOMF are still straight. Place GOMF on PAULA's socket, MAKE SURE ALL PINS LINE UP and the notch on PAULA faces the disk drive. Push straight down firmly and evenly. GOMF will ease itself into PAULA's socket. NOTE do not push sideways as you could very easily bend or break all the pins on GOMF.

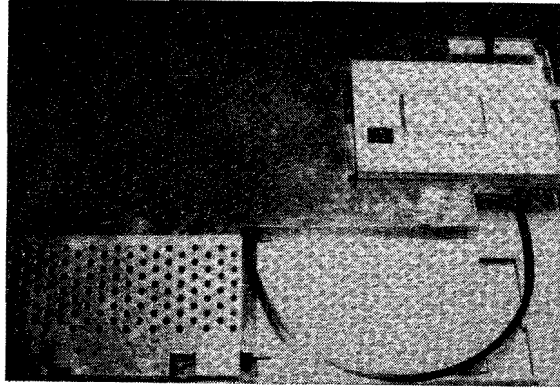


Picture 5

O) Refer to picture 6 for a suggested wire routing. Drill a 3/16 inch hole in the rear right hand corner of the case to mount the switch (you will most likely not be able to use the lock washer that comes with the switch). If you do not want to drill a hole in your case you could route the wire and switch out the expansion port hole.

P) To test the installation, temporarily re-connect the power supply, keyboard and monitor. When you turn the computer on, IF YOU DO NOT GET THE KICKSTART REQUESTER WITHIN THE NORMAL TIME DELAY then TURN THE COMPUTER OFF. Remove GOMF and re-install PAULA. Turn the power on, if everything appears normal, there may be a problem with the GOMF circuit. If the computer still doesn't function properly, take it to your dealer or computer repair center and have it checked. If you suspect a problem with the GOMF circuit, insert it into it's foam, box it, and send it to us, we will repair or replace

it and return it to you as soon as possible.



Picture 6

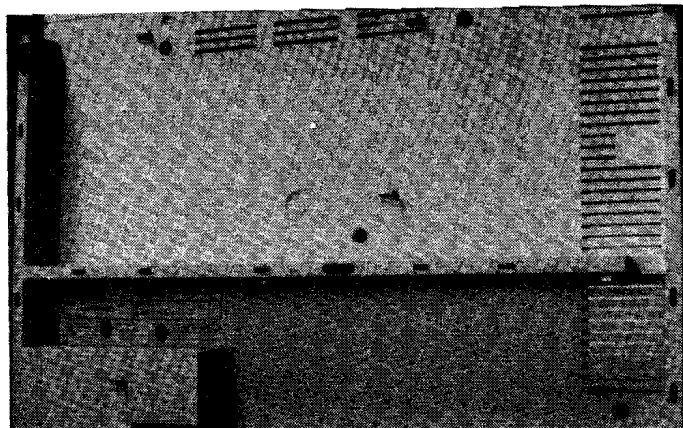
R) To reassemble the computer follow steps G through A.

S) Follow the instructions in the GOMF manual ERROR section to test the functions of the GOMF BUTTON.

AMIGA 1000 INSTALLATION:

A) Disconnect power cord, monitor, printer and any other accessories. If you have a memory expansion in the front of the computer, remove it now.

B) Refer to picture 7. Turn the computer upside down and locate the 5 phillips head screws that hold the case together and remove them. There are two more flat head phillips screws near the front edge of the computer, remove them also. Put them all to one side and identify them as case screws.



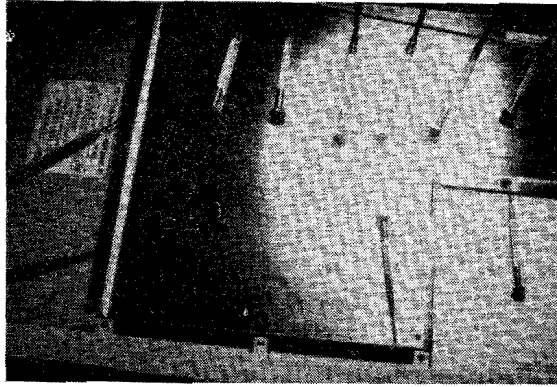
Picture 7

C) Turn the computer right side up and facing you. Remove the memory expansion cover (would have been removed when you took out the memory). There are a couple of plastic latches inside the case holding the top on. By gently wiggling the top of the case it should pop loose, do not force it. After the top is off the front will pull off too.

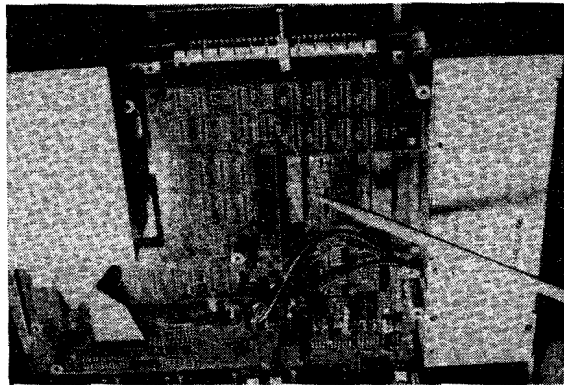
D) Refer to picture 8. There are numerous screws and 2 metal tabs holding the shield on. Remove the screws and identify them and straighten the 2 metal tabs with needle nose pliers. one tab is located at the right side of the shield and the other is at the rear left. Remove the shield by lifting up.

E) Refer to picture 9. Identify PAULA, it is near the corner of the daughter board, (see pointer in picture) the number on the I.C. will be 8364R?. The number after the R is a revision number and will vary depending on the date of manufacture.

F) An I.C. extractor may not fit in this location. Using a small slotted screwdriver, insert it between PAULA and the socket. By prying and twisting the screwdriver, lift PAULA evenly from each end.

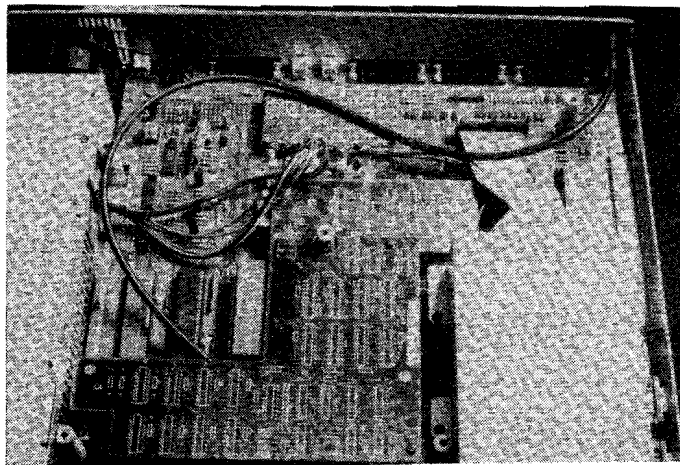


Picture 8



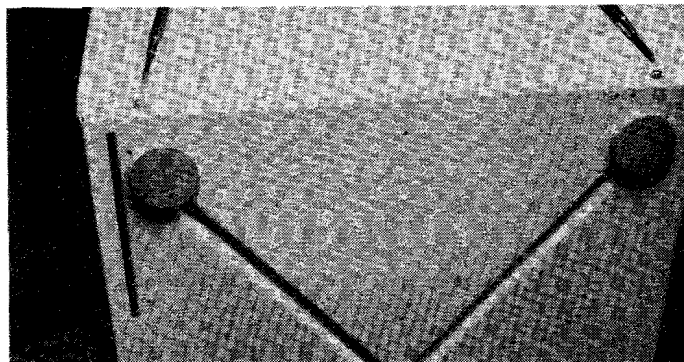
Picture 9

G) refer to AMIGA 500 instructions and follow J,K,L and M then continue here.



Picture 10

H) Refer to picture 10. Align GOMF on PAULA's socket with PAULA's notch facing the front of the computer. Make sure all pins line up with PAULA's socket. Push firmly STRAIGHT down and GOMF will ease into PAULA's socket.



Picture 11

I) Picture 10 shows a suggested BUTTON location in the top right corner of the back panel. On the right hand side of the shield there are some air ventilation holes. By enlarging one of the holes you will be able to route the wire and switch.

J) Refer to AMIGA 500 instructions and follow step P, then continue here.

K) To reassemble the computer follow steps D through A.

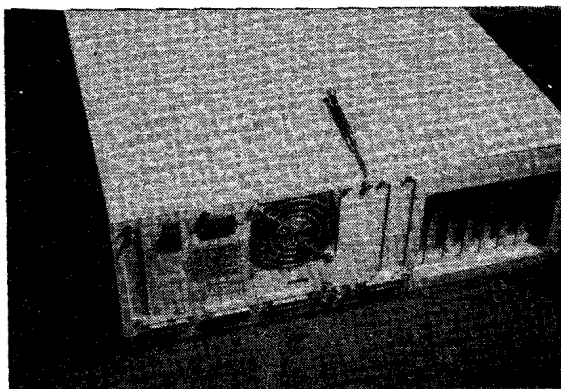
L) Refer to AMIGA 500 instructions and follow step S to complete testing.

AMIGA 2000 INSTRUCTIONS:

A) Disconnect power cord, monitor, printer and any other accessories.

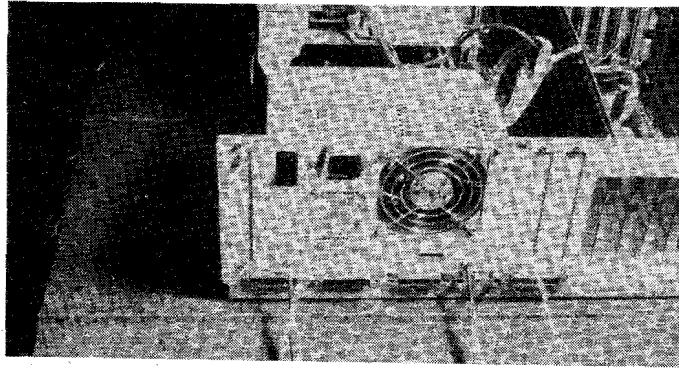
B) Refer to picture 11. remove the 2 screws from each side of the case.

C) Refer to picture 12. Remove the screw indicated. Set these 5 screws aside and identify them as case screws.



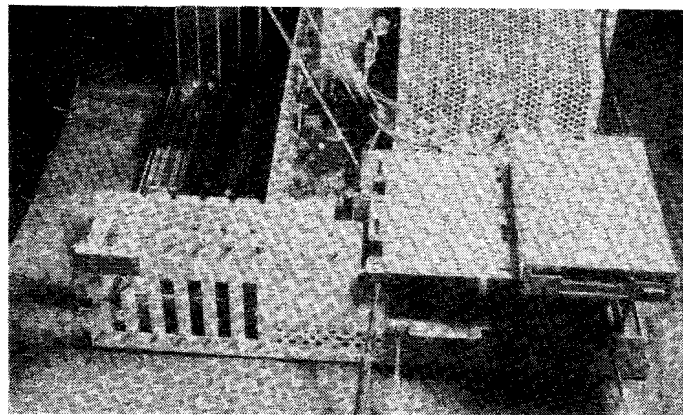
Picture 12

D) Refer to picture 13. Remove the 4 screws indicated that fasten the rear of the power supply.



Picture 13

E) Refer to picture 14. Remove the 3 screws indicated that fasten the front of the power supply. You should have 7 screws, identify them as power supply screws. NOTE the power supply won't fall down as it sits on a ledge.

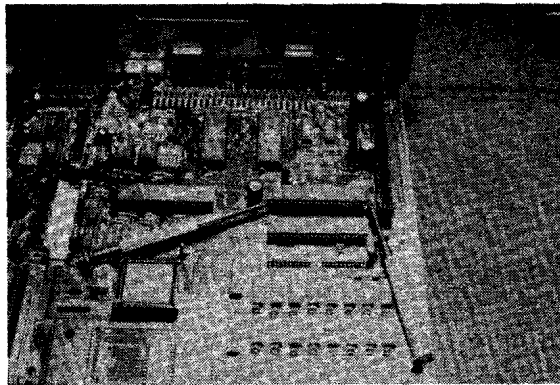


Picture 14

F) Looking from the front of the computer. You will notice a ribbon cable connecting to the main P.C.B. just to the left of the power supply. Note the direction that the ribbon cable exits the connector. Disconnect by pulling the connector straight up.

G) You will also notice a bundle of wires going to a connector ahead of the one you have just removed. Note the color of the wire at the front end of the connector before removing this connector also. You can now lift the power supply out of the computer and set it aside.

H) Refer to picture 15. identify PAULA, screwdrivers in picture point to PAULA. The number on the I.C. will be 8364R?. The number after the R is a revision number and will vary depending on date of manufacture. Note the direction of the notch on PAULA.

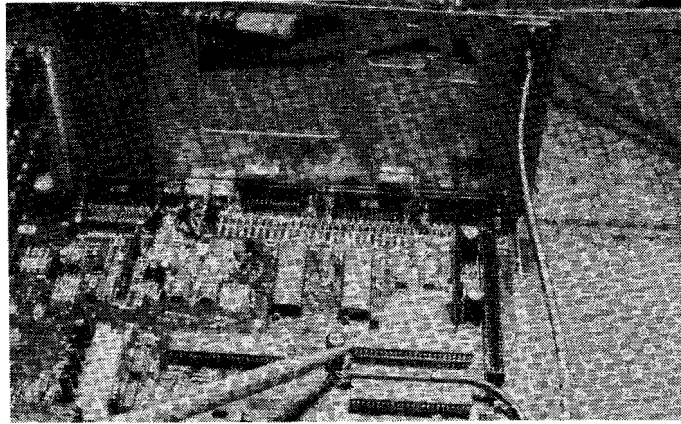


Picture 15

I) Follow steps I,J,K,L and M in the AMIGA 500 installation section before returning here.

J) Refer to picture 16. Make sure the pins on GOMF are still straight. Place GOMF on PAULA's socket, MAKE SURE ALL PINS LINE UP and the notch on PAULA faces the same way as you took it out. Push

straight down firmly and evenly. GOMF will ease into PAULA's socket.
NOTE do not push sideways as you could bend or break all of GOMF's pins.



Picture 16

K) Picture 16 shows a suggested cable routing and button location. Remove the metal cover and drill a 3/16 inch hole in it to mount the switch. You may not be able to use the lock washer supplied with the switch, if not then discard the lock washer.

L) proceed to step P in the AMIGA 500 installation instructions to test the installation.

M) To reassemble the computer follow steps G through A.

N) Follow the instructions in the GOMF manual ERROR section to test the functions of the GOMF BUTTON.

May all your GURU's be GOMFed, we wish you many happy hours of computing.

GLOSSARY

AMIGA: Wonder computer of the 80's. Tends to crash a bit, though.

AmigaDOS: Multitasking Operating System of the Amiga, written by Metacomco.

AMIGA-N & M keys: Method of switching between multiple screens. Hold left Amiga key (C= on A500) and press N or M

BOOT: One of GOMF3.0's gadgets. Causes the Amiga to RESET. (All programs in memory will be lost.)

BUGS: Insidious software errors that drive programmers mad.

CLI: Command Line Interpreter. Method of typing commands to access AmigaDOS functions.

CONMAN: PD utility allowing recall of previous CLI commands, hot keys, etc.

CRASH: Catastrophic system failure, see also GURU.

DECIGEL: Software patch for use with 68010. Traps certain incompatible 68000-specific codes. Use GOMF instead.

ERR1, ERR2, ERR3, ERR4: Software 'bug' examples included on GOMF3.0 disk.

EXCEPTION VECTORS: Motorola 68000 standard compatible processor vectors in table form. Do not trash these please.

EXEC MANUAL: One of the AMIGA developers' manuals. Deals mainly with 'KICKSTART'.

FPCP: Floating Point CoProcessor

FRED FISH: Fairly famous, foremost finder of fine floppies for free.

GADGETS: Clickable menu or window items (text or graphics).

GURU MEDITATION ALERT: You all know what THAT is, right?

HEY!: Early version of RECALL, distributed with GOMF1.0.

HOT KEY: A programmed function or macro that invokes a series of commands, etc. with a keystroke.

INTUITION: The Windows/Icons/Mouse/Pointer user interface of Amiga's Workbench. See also WIMP.

MC68000: Motorola microprocessor found in standard Amigas.

MC68010: Motorola microprocessor pin-compatible with 68000, average 5-10% faster.

MC68020: Powerful Motorola microprocessor.

NUKE: One of GOMF3.0's commands, allows the user to remove tasks from the system.

PD: Public domain, a freely distributable program. See Appendix B.

PMMU: Paged Memory Management Unit

POPCLI: PD program to call a new CLI by pressing ALT-ESC, also blanks screen when system is idle.

PROCESS: A program, usually invoked from Workbench.

RAM: Amiga's Ramdisk device.

RECALL: Command to re-invoke GOMF3.0

SCAT: Gadget that causes GOMF3.0's window to jump to the

opposite corner of the screen it is on.

SCREEN: The kind of Amiga display that can be dragged the whole width of the screen up or down. The Workbench screen is an example of a typical SCREEN.

SHAREWARE: What GOMF1.0 is, and GOMF3.0 isn't. In an ideal world, programs voluntarily paid for by the people that use 'em.

SOFTWARE: A place where bugs live.

STACK: The processor's workspace when running any task.

STARTUP-SEQUENCE: List of commands in S directory automatically executed on startup.

SUPERVISOR STACK FRAME: The system's private stack holds this predefined set of data when in its special supervisor mode.

TASK: A program, usually invoked by a CLI command.

VD0: Recoverable Ramdisk. Shareware by Perry Kivolowitz of ASDG

VECTOR TABLE: A table of pointers to various important system ROM points.

WHAP: A gadget that allows GOMF to remove windows or screens belonging to other programs. Note that windows can be WHAPPED anywhere inside their perimeters, but to WHAP a screen you must click on its FRONT GADGET.

WIMP: Windows/Icons/Mouse/Pointer

WINDOW: A display component on a SCREEN. When you double-click on a disk icon you will open a WINDOW on the Work-

COPYRIGHT: The program and its related documentation are copyrighted.

LICENCE: You have the non-exclusive right to use the enclosed program only on a single computer. You may physically transfer the program from one computer to another provided that the program is used on only one computer at a time. You may not electronically transfer the program from one computer to another over a network. You may not distribute copies of the program or accompanying documentation to others. You may not modify or translate the program or documentation.

TERMS: This licence is effective until terminated. You may terminate it by destroying the program and documentation and all copies thereof. This licence will also terminate if you fail to comply with any term or condition of this agreement. You agree upon such termination to destroy all copies of the program and documentation.

DISCLAIMER OF ALL SOFTWARE WARRANTIES AND LIABILITY

Hypertek/Silicon Springs Development Corp. provides this program "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the results and performance of the program is assumed by you. Should the program prove defective, you (and not Hypertek/Silicon Springs Development Corp. or its dealers) assume the entire cost of all necessary servicing, repair or correction. Further, Hypertek/Silicon Springs Development Corp. does not warrant, guarantee or make any representation regarding the use of, or the results of the use of, the program in terms of correctness, accuracy, reliability, currentness, or otherwise; and you rely on the program and results solely at your own risk.

DISKETTE LIMITED WARRANTY

Hypertek/Silicon Springs Development Corp. warrants to the original licensee that the diskette on which the program is recorded shall be free

from defects in material and workmanship only for a period of thirty (30) days from the date of original purchase. If a defect covered by this warranty occurs during this 30-day warranty period, and it is returned to the dealer from whom it was purchased not later than five (5) days after the end of such 30-day period, the dealer shall, at the dealer's option, either repair or replace the diskette.

This warranty is in lieu of all other express or implied warranties, and the duration of any implied warranty, including but not limited to the implied warranties of merchantability and fitness for a particular purpose, is hereby limited to said thirty (30) day period. Hypertek/Silicon Springs Development Corp.'s liability is limited solely to the repair or replacement of the defective product, in its sole discretion, and shall not in any event include damages for loss of use or loss of revenues, anticipated costs, expenses or damages, including without limitation any data or information which may be lost or rendered inaccurate, even if Hypertek/Silicon Springs Development Corp. has been advised of the possibility of such damages.

Some states do not allow a limitation on how long an implied warranty lasts, so the above limitation may not apply to you. Some states do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may have other rights which vary from state to state.

HARDWARE DISCLAIMER

Hypertek/Silicon Springs Development Corp. provides this hardware "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the results and performance of the hardware is assumed by you. Should the hardware prove defective, you (and not Hypertek/Silicon Springs Development Corp. or its dealers) assume the entire cost of all necessary servicing, repair or correction. Further, Hypertek/Silicon Springs Development Corp. does not warrant, guarantee or make any representation regarding the use of, or the results of the use of, the hardware in terms of

effectiveness, precision, accuracy, reliability, currentness, or otherwise; and you rely on the hardware and results solely at your own risk.

HARDWARE LIMITED WARRANTY

Hypertek/Silicon Springs Development Corp. warrants to the original licensee that the hardware device (which may or may not have been included in this package) shall be free from defects in material and workmanship only for a period of thirty (30) days from the date of original purchase. If a defect covered by this warranty occurs during this 30-day warranty period, and it is returned to the dealer from whom it was purchased not later than five (5) days after the end of such 30-day period, the dealer shall, at the dealer's option, either repair or replace the hardware.

This warranty is in lieu of all other express or implied warranties, and the duration of any implied warranty, including but not limited to the implied warranties of merchantability and fitness for a particular purpose, is hereby limited to said thirty (30) day period. Hypertek/Silicon Springs Development Corp.'s liability is limited solely to the repair or replacement of the defective product, in its sole discretion, and shall not in any event include damages to computers or any other connected hardware device or interface, loss of use or loss of revenues, anticipated costs, expenses or damages, including without limitation any data or information which may be lost or rendered inaccurate, even if Hypertek/Silicon Springs Development Corp. has been advised of the possibility of such damages.

NOTICE:

Hypertek/Silicon Springs Development Corp. reserves the right to make improvements to the program described herein and related documentation at any time and without notice.

Hypertek/Silicon Springs Development Corp.
812 Surrey Street
New Westminster, B.C. Canada V3L 4W2
(604) 524-1125

INDEX

68010/20 (18), (22)

Advanced Applications (77)

Advanced Tutorial (72)

ALINK (77)

AMICUS (91)

AmigaDOS task array (40), (43)

APTR (78)

Background tasks (13)

BOOT (37), (57)

BootStrap (89)

Branching Errors (39)

Bridgecard (17)

CANCEL (43), (68)

CLI (13), (14)

How To Use (60)

Commands

BindDrivers (17)

CD (61), (62)

CTRL-D (79)

DEBUG (79)

DJMount (18)

ENDCLI (79)

LOADWB (79)

RAM: (9)

RECALL (45), (70)

RUN (18), (25)

SetClock (37), (57)

STACK (16)

Wait (37), (57)

CONMAN (50), (92)

Crashes

surviving (21)

Credits (82)

- DEBUG (79)
- DeciGel (22), (91)
- Devices (85)
 - NIL: (18), (25)
 - RAM: (9)
- Disk Resource (86)
- DOS Library (85)

- ENDCLI (79)
- Error Types (30)
 - GURU Meditation (30)
 - Lockups (31)
 - Task Held (30)
- Error1 (47),(54),(12)
- Error2 (47),(58),(12)
- Error3 (47),(50),(12)
- Error4 (47),(60),(12)
- Error5 (47),(71),(12)
- Error6 (12),(71)
- Errors
 - Branching (39)
 - Drawer (47)
 - During NUKE (73)
 - During RECALL (73)
 - Frequent (80)
 - Functional (39)
 - Hardware (80)
 - Lockups (34), (71)
 - Multiple (72), (73)
 - Skipping over (21), (33)
 - Task Held (56,(30),(31)
 - That can't SKIP (34)
- Exec Library (85)
- Exec Ready Queue (40)
- Exec Task Ready Queue (43)
- Exec Waiting Queue (40)
- Expansion Library (85)
- Extended selection (42),(66)

Format of Guru Meditation Messages (85)

FPCP (90),(107)

FRED FISH (91)

Functional Errors (39)

Gadgets

BOOT (37), (57)

CANCEL (43), (68)

GOMF (34), (57)

GURU (34), (57)

LAST (43), (67)

NEXT (43), (67)

NUKE (39), (43), (67)

RETRY (43), (68)

SCAT (37), (57)

Task Fields (43), (67)

WHAP (36)

General Error Codes (85)

GOMF (34)

“^” Function (18)

‘&’ function (18)

BUTTON (21)

CLI Startup (25)

Errors (47)

Installing (15)

Installing on Hard Disk (17)

Nuke’ing (44), (69)

Reentering (52), (69)

Shareware (7),(10)

Startup Display (29)

Testing Functionality (45), (70)

Updates (22)

Upgrading (19)

What it does (25)

With a 68010/20 (18)

Workbench (26), (28)

Workbench 1.2 (15)

Workbench 1.3 (15)

GOMF Button

Practising with (70)
Skipping over Errors (21),(33)
Thawing System Freezes (35)
GOMF1.0
 link module (11)
Gomf1.0.obj (77)
Graphics Library (85)
GURU (37), (57)

Hard Disk
 Resetting (37),(57)
Hardware
 Resetting (37),(57)
HARDWARE INSTALLATION (93)

Interlace (37)
Intuition Library (85)

Kickstart (79),(80)

LAST (43), (67)
Layers Library (85)
LEADER BOARD GOLF
 (80)
LexCheck (69), (45)
Libraries (85)
LOADWB (79)
Lockup (21)
Lockups
 Unlocking (34),(71)

MARAUDER (69), (45)
MEMORY
 Adding (81)
MemWatch (92)
MOREROWS (22)
MOVE SR,<ea> (47),(18)
Multiple errors (72),(73)
MUSICRAFT (80)

- Network
 - Resetting (37),(57)
- NEXT (43), (67)
- NUKE (39), (42), (43), (66),(67)
 - "^" (40)
 - "!" (40)
 - Description (39)
 - Errors During (73)
 - From the CLI (39)
 - From the Workbench (42), (66)
 - Gadgets (39),(43),(67)
 - Parsing (41)
 - Syntax (40)
 - TASK SELECTOR (42)
 - Two Programs (44), (68)
- Nuke'ing
 - Error4 (63)
 - GOMF (44), (69)
 - Two tasks (44), (68)
- PAL Video (22)
- PC (49)
- PMMU (108)
- POPCLI (9), (92)
- Pre-Panic Checklist (79)
- Privilege (19)
- Program Counter (PC) (34)
- Public Domain Software (91)
- Quick Startup (15)
- RAM: (9)
- Ram Disk (9)
- RAM Library (88)
- Re-entrancy (72)
- Real time clock (37)
- RECALL (45), (52), (70)
 - Errors During (73)
- Resources (86)

RETRY (43), (68)

RomWACK (91)

RUNBACK (15)

RUN (18),(25)

SCAT (37), (57)

Screens

 Removing (36)

SetClock (37),(57)

SETPATCH (22)

Specific Alert Codes (86)

STACK command (16)

StarGlider (69)

Startup-sequence (13), (16)

Subsystem ID codes (85)

Task Fields (43), (67)

TASK HELD (56),(30),(31)

Task Ready Queue (65)

Task Waiting Queue (43)

Timer Device (85)

TOOLTYPES

 PRIVILEGE (27)

 VECTORS (27)

TrackDisk Device (85)

TRANSFORMER (80)

Tricks & Tips(79)

TroubleShooting (79)

Tutorial

 Beginner's (51)

 Advanced (72)

UBYTE (78)

ULONG (78)

VD0: (9)

Vectors (18), (26), (45), (69)

Wait (37),(57)

WHAP (36), (56)

Windows

Removing (36)

Workbench (13)

Processes (14)

1. *Pharmaceutical Innovation and the Role of Government*
 2. *The Impact of Patent Law on Drug Development*
 3. *The Role of Government in Regulating Pharmaceuticals*
 4. *The Impact of Health Insurance on Pharmaceutical Sales*
 5. *The Role of Government in Funding Pharmaceutical Research*
 6. *The Impact of Drug Pricing on Access to Medicine*
 7. *The Role of Government in Controlling Drug Quality*
 8. *The Impact of Drug Safety on Pharmaceutical Sales*
 9. *The Role of Government in Promoting Public Health*
 10. *The Impact of Drug Abuse on Pharmaceutical Sales*
 11. *The Role of Government in Controlling Drug Trafficking*
 12. *The Impact of Drug Dependence on Pharmaceutical Sales*
 13. *The Role of Government in Promoting Mental Health*
 14. *The Impact of Drug Misuse on Pharmaceutical Sales*
 15. *The Role of Government in Controlling Drug Production*
 16. *The Impact of Drug Diversion on Pharmaceutical Sales*
 17. *The Role of Government in Promoting Substance Abuse Prevention*
 18. *The Impact of Drug Overdose on Pharmaceutical Sales*
 19. *The Role of Government in Controlling Drug Distribution*
 20. *The Impact of Drug Resistance on Pharmaceutical Sales*
 21. *The Role of Government in Promoting Antimicrobial Stewardship*
 22. *The Impact of Drug Resistance on Pharmaceutical Sales*
 23. *The Role of Government in Controlling Drug Resistance*
 24. *The Impact of Drug Resistance on Pharmaceutical Sales*
 25. *The Role of Government in Promoting Drug Resistance Prevention*
 26. *The Impact of Drug Resistance on Pharmaceutical Sales*
 27. *The Role of Government in Controlling Drug Resistance*
 28. *The Impact of Drug Resistance on Pharmaceutical Sales*
 29. *The Role of Government in Promoting Drug Resistance Prevention*
 30. *The Impact of Drug Resistance on Pharmaceutical Sales*
 31. *The Role of Government in Controlling Drug Resistance*
 32. *The Impact of Drug Resistance on Pharmaceutical Sales*
 33. *The Role of Government in Promoting Drug Resistance Prevention*
 34. *The Impact of Drug Resistance on Pharmaceutical Sales*
 35. *The Role of Government in Controlling Drug Resistance*
 36. *The Impact of Drug Resistance on Pharmaceutical Sales*
 37. *The Role of Government in Promoting Drug Resistance Prevention*
 38. *The Impact of Drug Resistance on Pharmaceutical Sales*
 39. *The Role of Government in Controlling Drug Resistance*
 40. *The Impact of Drug Resistance on Pharmaceutical Sales*
 41. *The Role of Government in Promoting Drug Resistance Prevention*
 42. *The Impact of Drug Resistance on Pharmaceutical Sales*
 43. *The Role of Government in Controlling Drug Resistance*
 44. *The Impact of Drug Resistance on Pharmaceutical Sales*
 45. *The Role of Government in Promoting Drug Resistance Prevention*
 46. *The Impact of Drug Resistance on Pharmaceutical Sales*
 47. *The Role of Government in Controlling Drug Resistance*
 48. *The Impact of Drug Resistance on Pharmaceutical Sales*
 49. *The Role of Government in Promoting Drug Resistance Prevention*
 50. *The Impact of Drug Resistance on Pharmaceutical Sales*
 51. *The Role of Government in Controlling Drug Resistance*
 52. *The Impact of Drug Resistance on Pharmaceutical Sales*
 53. *The Role of Government in Promoting Drug Resistance Prevention*
 54. *The Impact of Drug Resistance on Pharmaceutical Sales*
 55. *The Role of Government in Controlling Drug Resistance*
 56. *The Impact of Drug Resistance on Pharmaceutical Sales*
 57. *The Role of Government in Promoting Drug Resistance Prevention*
 58. *The Impact of Drug Resistance on Pharmaceutical Sales*
 59. *The Role of Government in Controlling Drug Resistance*
 60. *The Impact of Drug Resistance on Pharmaceutical Sales*
 61. *The Role of Government in Promoting Drug Resistance Prevention*
 62. *The Impact of Drug Resistance on Pharmaceutical Sales*
 63. *The Role of Government in Controlling Drug Resistance*
 64. *The Impact of Drug Resistance on Pharmaceutical Sales*
 65. *The Role of Government in Promoting Drug Resistance Prevention*
 66. *The Impact of Drug Resistance on Pharmaceutical Sales*
 67. *The Role of Government in Controlling Drug Resistance*
 68. *The Impact of Drug Resistance on Pharmaceutical Sales*
 69. *The Role of Government in Promoting Drug Resistance Prevention*
 70. *The Impact of Drug Resistance on Pharmaceutical Sales*
 71. *The Role of Government in Controlling Drug Resistance*
 72. *The Impact of Drug Resistance on Pharmaceutical Sales*
 73. *The Role of Government in Promoting Drug Resistance Prevention*
 74. *The Impact of Drug Resistance on Pharmaceutical Sales*
 75. *The Role of Government in Controlling Drug Resistance*
 76. *The Impact of Drug Resistance on Pharmaceutical Sales*
 77. *The Role of Government in Promoting Drug Resistance Prevention*
 78. *The Impact of Drug Resistance on Pharmaceutical Sales*
 79. *The Role of Government in Controlling Drug Resistance*
 80. *The Impact of Drug Resistance on Pharmaceutical Sales*
 81. *The Role of Government in Promoting Drug Resistance Prevention*
 82. *The Impact of Drug Resistance on Pharmaceutical Sales*
 83. *The Role of Government in Controlling Drug Resistance*
 84. *The Impact of Drug Resistance on Pharmaceutical Sales*
 85. *The Role of Government in Promoting Drug Resistance Prevention*
 86. *The Impact of Drug Resistance on Pharmaceutical Sales*
 87. *The Role of Government in Controlling Drug Resistance*
 88. *The Impact of Drug Resistance on Pharmaceutical Sales*
 89. *The Role of Government in Promoting Drug Resistance Prevention*
 90. *The Impact of Drug Resistance on Pharmaceutical Sales*
 91. *The Role of Government in Controlling Drug Resistance*
 92. *The Impact of Drug Resistance on Pharmaceutical Sales*
 93. *The Role of Government in Promoting Drug Resistance Prevention*
 94. *The Impact of Drug Resistance on Pharmaceutical Sales*
 95. *The Role of Government in Controlling Drug Resistance*
 96. *The Impact of Drug Resistance on Pharmaceutical Sales*
 97. *The Role of Government in Promoting Drug Resistance Prevention*
 98. *The Impact of Drug Resistance on Pharmaceutical Sales*
 99. *The Role of Government in Controlling Drug Resistance*
 100. *The Impact of Drug Resistance on Pharmaceutical Sales*

